

INPUT
Publicación práctica para usuarios de
sinclair
Revista mensual 1986
Precio 350 Ptas
Año 1 Número 10
RECETARIO DE CODIGO MAQUINA
NO TE CONFORMES CON 21 GDU
MODELANDO LA REALIDAD



AÑO 1 NUMERO 10

DIRECTOR:
Alejandro Diges
COORDINADOR EDITORIAL:
Francisco de Molina
DISEÑO GRAFICO:
Tomás López
COLABORADORES:
Antonio Taratiel, Luis R. Palencia, Francisco Tórtola, Benito Román, Esther de la Cal, Ernesto del Valle, Equipo Molisoft.

INPUT Sinclair es una publicación juvenil de EDICIONES FORUM

GERENTE DIVISION DE REVISTAS:
Angel Sabat

PUBLICIDAD: José Real-Grupo Jota
Madrid: c/ General Varela, 35
Teléf. 270 47 02/03
Barcelona: Avda. de Sarriá, 11-13, 1.º
Teléf. 250 23 99

FOTOMECANICA: Ochoa, S. A.

COMPOSICION: EFCA, S. A.

IMPRESION: Sirven Grafic
C/ Gran Vía, 754-756. 08013 Barcelona
Depósito legal: M. 27.884-1985

SUSCRIPCIONES: EDISA,
López de Hoyos, 141. 28002 Madrid
Teléf. (91) 415 97 12

REDACCION:
Alberto Alcocer, 46, 4.º
28016 Madrid. Teléf. 250 10 00

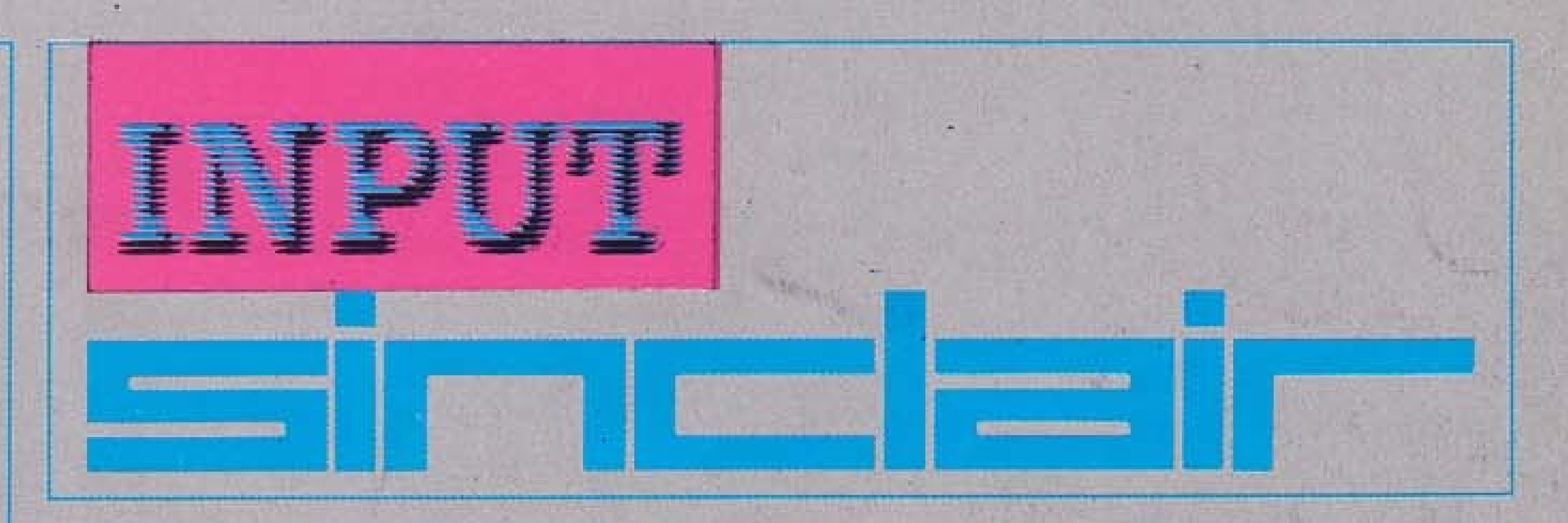
DISTRIBUIDORA
R.B.A. PROMOTORA DE EDICIONES, S. A.
Travesera de Gracia, 56. Edificio Odiseus.
08006 Barcelona.

El precio será el mismo para Canarias que para la Península y en él irá incluida la sobretasa aérea.

Se ha solicitado el control OJD

INPUT Sinclair es independiente y no está vinculada a Sinclair Research o sus distribuidores.

INPUT no mantiene correspondencia con sus lectores, si bien la recibe, no responsabilizándose de su pérdida o extravío. Las respuestas se canalizarán a través de las secciones adecuadas en estas páginas.



SUMARIO



TANTO MONTA...

Era de esperar que la sorprendente absorción de **Sinclair** a manos del estratega del *marketing* británico, **Alan Sugar**, generase toda una cascada de noticias. Sin embargo tanto los representantes españoles de **Sinclair** como de **Amstrad** no han dicho «esta boca es mía», dejándonos con las ganas de saber qué ocurrirá.

Un paseo por el Reino Unido, donde los canales de comunicación suelen ser más ágiles, podría servir para indagar en torno a un futuro poco aclarado. Saltamos el Canal decididos a preguntar a quien hiciera falta.

Por lo pronto, los **Spectrum 128** muestran en los escaparates de aquella isla una considerable bajada de precio. En nuestro país los expertos estiman que se habrán vendido unas tres mil unidades, el mismo orden de magnitud aplicable a los malogrados **QL**, que no parecen tener asegurada su descendencia.

La pregunta que surge es obvia: ¿Es ésta una maniobra destinada a la liquidación de los *stocks* acumulados de **Spectrum**? Las respuestas apuntan hacia una diversificación del mercado por el que peleaban **Sinclair** y **Amstrad.** Por un lado el **Spectrum,** lejos de desaparecer, estaría a punto de encontrar una continuidad a final de este año con un modelo de 128 K, que llevaría un *cassette* incorporado en la misma carcasa, convirtiéndose en el nuevo rey del ordenador doméstico con mucho *software* de juegos. Tampoco se descarta la utilización del *microdiskette* de 3 pulgadas. Un precio de partida barajado para el futuro **Sinclair** es de 139 libras esterlinas (algo más de 30.000 pesetas).

Por otro lado, **Amstrad** se reorientaría al más lucrativo segmento del ordenador utilizable por profesionales y el tratamiento de textos, del que son buenos ejemplos el **PCW 8256** y el **8512.**

También se especula mucho en torno a un posible compatible **PC** de **IBM,** producido por **Amstrad.**

Cambiando de terreno, debemos reconocer que la picaresca y las ganas de broma siguen siendo uno de los rasgos característicos de nuestro país. Pues bien, hemos recibido algunas llamadas de lectores que habéis recibido una carta (eso sí, llena de tachaduras) en las que se comunica la concesión de un hipotético premio. Los premiados en INPUT aparecen únicamente en las listas publicadas en la revista y reciben posteriormente una carta convenientemente firmada, que incorpora el logotipo de la Editorial (no el de la revista). Así que, atentos a las bromitas.

LOS MEJORES DE INPUT

Hemos pensado que es interesante disponer de un ***ranking*** que ponga en claro, mes a mes, cuáles son los programas preferidos de nuestros lectores. Para ello, es obligado preguntaros directamente y tener así el mejor termómetro para conocer vuestras preferencias. Podéis votar por cualquier programa aunque no haya sido comentado todavía en **INPUT**.

El resultado de las votaciones será publicado en cada número de **INPUT**.

Entre los votantes sortearemos 10 cintas de los títulos que pidáis en vuestros cupones.

Nota: No es preciso que cortéis la revista, una copia hecha a máquina o una simple fotocopia sirven.

Enviad vuestros votos a: **LOS MEJORES DE INPUT** Alberto Alcocer, 46 - 4.° B. 28016 Madrid

ELIGE TUS PROGRAMAS

Primer título elegido ____________________

Segundo título elegido ____________________

Tercer título elegido ____________________

Programa que te gustaría conseguir ____________________

Qué ordenador tienes ____________________

Nombre ____________________

1.er Apellido ____________________

2.° Apellido ____________________

Fecha de nacimiento ____________________

Teléfono ____________________

Dirección ____________________

Localidad ____________________

Provincia ____________________



LOS GENIOS NUNCA MUEREN

Pasados los primeros momentos de incertidumbre, las aguas vuelven a su cauce. Uno de los efectos mas dramáticos de la venta de la sección informática de Sincalir Research a Amstrad se ha dejado notar sobre la plantilla de *sir* Clive, de tal manera que la anteriormente prospera compañia se ha quedado "en cuadro". Sin embargo, pese a la profunda reestructuración, permanece buena parte de los antiguos proyectos. De hecho Sinclair continua siendo propietario del ochenta por ciento de su compañia y puede desarrollar, pero no poner en el mercado por si mismo, nuevos ordenadores. El misterioso Pandora, ordenador portátil con pantalla plana, podría incluso llegar a ver la luz de la mano comercializadora de Amstrad. Con respecto al Pandora se afirma que no estará disponible para su fabricación masiva hasta el año que viene y podría llevar una unidad de disco incorporada e incluso utilizar el sistema operativo CP/M.
El interés de *sir* Clive parecen continuar por los mismos derroteros: Los *chips* de grandisima escala de integración a nivel de oblea (plaqueta de silicio de mayor tamaño que un solo *chip*) y los radiotelefonos personales. Con el primer proyecto espera haber concentrado cuarenta millones de bits de almacenamiento en una oblea para el próximo año.
Por otro lado, el Spectrum 128 tiene asegurada su continuidad y parece probable que a finales de este año aparezca un nuevo modelo de Spectrum con la unidad de *cassette* (o disco de 3") incoporado en la misma carcasa, siendo destinado al segmento de los microordenadores para videojuegos y divertimento. Seguiremos atentos para informar.



NUEVO SOFT DE COMPULOGICAL

La firma distribuidora de *software* Compulogical también ha firmado nuevos contratos de representación exclusiva. Por un lado el acuerdo con Softech/Edge aporta los programas Quest of Mindstone, el espectacular ajedrez Looking glasschess y las utilidades The writer y Artist 2.
Por otro lado, la firma con Consolidated Software le hará disponer del titulo Pipeline.



NUEVO STRIP POKER

Samantha Fox es una modelo británica muy popular por sus apariciones en prensa portando poca (ninguna) ropa. La firma Martech la ha convencido para que preste su efigie y su experiencia con el naipe para un nuevo juego de *strip poker*, que hace furor entre los *espectrumeros* de aquel pais. El próximo mes comentaremos en juego en la sección correspondiente. Por el momento, nuestro experto comentarista ya ha conseguido desposeerla de los guantes. Podrá llegar mas lejos?...



SOFTWARE DE FIREBIRD PARA EL 128

Firebird es una firma filial de la "Telefónica" británica, British Telecom, que no para de lanzar nuevos titulos al mercado microinformático. Según un alto ejecutivo de la firma, a finales de mayo tiene previsto lanzar el esperado programa T.C.U.P.S. (este INPUT entró en máquinas a mediados del mes) en versiones 48 K y 128 K. Asimismo, se espera a comienzos de junio la aparición de las versiones de Elite y Rasputin para el controvertido 128, los cuales incorporan un mayor número de pantallas en base a la mayor disponibilidad de memoria. En este mismo mes lanzarán The Heartland, en versión 48 K, que parece albergar buenos auspicios.
Estos desarollos para el 128 muy bien podrían responder a las nuevas espectativas creadas por Amstrad, quien muestra mayor interés en potenciar este modelo que su anterior fabricante (al menos en el Reino Unido).



LOS FINALISTAS DEL MUNDIAL

U.S. Gold acaba de lanzar el programa World Cup Carnival, un juego de futbol capaz de trasladar la emoción de los mundiales a la pantalla del ordenador, enfrentando a los equipos elegidos por el usuario. Para darle mayor realismo, esta firma ha negociado la utilización de los emblemas y anagramas oficiales con la FIFA.
En la carpeta viene adjunto un cartel que permite seguir la evolución de los equipos, en el que se pegarán las banderas adhesivas de los paises no eliminados.
La firma hace un ofrecimiento a los lectores de INPUT. Entre quienes acierten los dos equipos que jugarán la final se sorteará un ganador, que será premiado con una copia de todos los juegos de U.S. Gold lanzados durante el mes de julio. A la carta, enviada a nuestra redacción, habrá que adjuntar una de las banderas adhesivas del equipo español, que vienen con el programa. La fecha tope de recepción es el próximo 19 de junio.

UN PROGRAMA QUE JUEGA A LAS CUATRO EN RAYA

Se trata de un programa que lleva a la práctica todos los conceptos que estudiamos cuando «desmenuzábamos» el juego de las «Cuatro en Raya». De paso, con este programa trataremos de ilustrar las dificultades que presenta el BASIC en esta materia, y sobre todo, cómo superarlas.

En el último artículo publicado dentro de esta serie, ilustrábamos los principios del funcionamiento de los juegos de estrategia con el ejemplo de las «Cuatro en Raya». Creemos que las explicaciones eran lo suficientemente detalladas como para que algún lector avezado se atreviera a desarrollar su propio juego inteligente. Tanto para los que lo han intentado como para los que no, este mes proponemos lo que podría ser la solución al reto que planteábamos en el último número.

Por si algún lector no conociera este juego, o tuviera alguna duda por su similitud con el de las «Tres en Raya», empezaremos por explicar someramente en qué consiste. Se trata de colocar cuatro fichas de un mismo color en línea, ya sea horizontal, vertical o diagonal. Cada ficha ha de colocarse sobre la base del tablero o encima de otra ficha ya colocada, aunque sea del contrario. El primer jugador que consiga su objetivo, será el ganador. Es fundamental tener en cuenta la importancia del hecho de «colocar primero» en el desarrollo posterior del juego. Por esta razón, el programa incopora la posibilidad de elegir quien comienza el juego, el ordenador o el usuario.

Volvemos a hacer hincapié en cómo se demuestra que la **Inteligencia Artificial** puede hacer cosas sorprendentes con listados sorprendentemente breves.

DESGLOSADO DEL PROGRAMA

Para una clara explicación del funcionamiento del programa, nos remitiremos constantemente al listado del mismo y a los gráficos ilustrativos que acompañan al texto.

Como vimos en números pasados, el fundamento básico de los juegos de estrategia está en la valoración de las jugadas posibles, propias y del contrario, para determinar cuál es la más favorable y actuar en consecuencia. Pués bien, en este caso, como en todos, deberemos empezar por establecer cuáles son las posiciones en las que podemos colocar una ficha. La figura número uno lo muestra (conociendo las reglas del juego, no necesita más comentario). Nosotros podemos verlo a simple vista, pero ¿cómo hacerselo ver al ordenador?

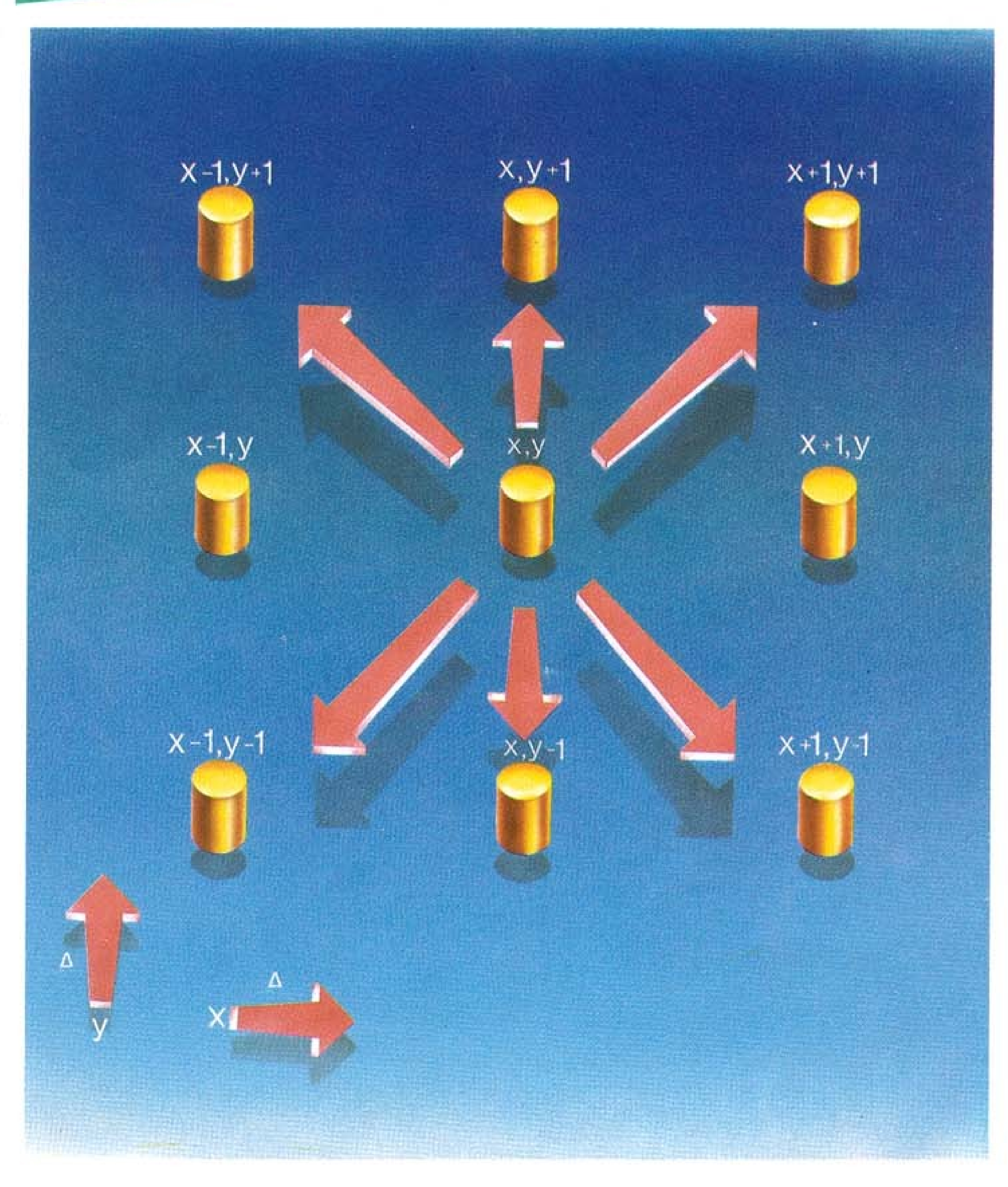
De esto se encarga la rutina-bucle situada en la línea 1030-1060.

El tablero está representado en la matriz a(8,5), con un cero para las posiciones vacías, un uno para las ocupadas por una ficha del ordenador, y un dos para las ocupadas por el jugador. El bucle 1030-1060 comienza a buscar posiciones vacías desde la base del tablero, enviando el flujo del programa a la parte siguiente (valoración de la posición) cuando encuentre una. Para ver la relación tablero/matriz (que está pensada a la inversa de como cabría esperar para simplificar el proceso), consultar la figura cuatro.

A partir de aquí, posición por posición, el programa estudia y valora los alineamientos conseguidos. Almacena el valor más alto conseguido en la variable tff, opera de la misma manera con las posibles jugadas del contrario, y finalmente hace la jugada más favorable. Si la mejor jugada del ordenador es un alineamiento de dos fichas, y la mejor jugada posible del contrario otro de tres, el ordenador hará la jugada del contrario para evitar que éste la realice, pues su valor (tres) es mayor. El programa puntúa con un

Figura 1.- Los lugares a analizar en la presente situación aparecen coloreados en magenta.

Figura 2.- Análisis de una posición con respecto a las posiciones contiguas.



punto cada ficha alineada en una posición dada. Este análisis y valoración configura lo que podemos considerar como la parte más importante del programa, que se encuentra entre las líneas 1100-1530. En cada posición analizada, se estudian las alineaciones conseguidas en la forma que muestra la figura número dos. Se comienza por analizar una posición contigua; por ejemplo, x+1,y−1. Si está ocupada por una ficha del mismo color (representada en la matriz por un 1 si es del ordenador, y por un 2 si es del contrario), se suma un punto al valor de esa posición, que comenzó con cero, y se vuelve a sumar uno a 'x' y a restar uno a 'y' para seguir la línea en esa dirección, es decir, se analiza la posición x+2, y−2. Si no está ocupada por una ficha del mismo color, se vuelve a la posición original (x,y), y se pasa a seguir en el sentido contrario dentro de la misma línea (es decir, x−1, y+1), procediendo de la misma manera. Al final obtendremos el número de fichas alineadas y contiguas en la recta representada por las dos flechas verdes. Sucesivamente, se hace un análisis idéntico con el resto de las rectas (la horizontal, vertical y la otra oblicua) y se almacena, como ya dijimos, el mayor alineamiento en la variable tff.

Quizás esta explicación resulte un poco complicada si se hace con palabras. La mejor forma de comprender esto que hemos expuesto es seguir detenidamente el listado del programa, para estudiar directamente la forma en que lo hace.

Vamos a tomar ahora el ejemplo representado en la figura tres. El ordenador estudia y valora dos jugadas propias posibles, que se realizarían colocando en la posición 1 o en la posición 2. En la posición uno, obtendría un alineamiento de dos fichas, con un valor pues de 1 punto (la ficha que aún no se ha colocado no se cuenta, por eso es uno y no dos), mientras que en la posición 2, el alineamiento sería de 3 fichas, con un valor de 2. La posición uno, por tener un valor menor, queda descartada. A continuación, se busca la mejor jugada del contrario (impropiamente, decimos «mejor jugada» cuando deberíamos decir «alineamiento más numeroso», ya que en realidad, un alineamiento de tres no tienen por qué ser necesariamente más beneficioso que uno de dos en la estrategia general del jugador).

Colocando éste en la posición tres, obtendría un alineamiento de cuatro fichas, y por tanto vencería. En todo caso, como su valor es mayor que el obtenido por el mejor alineamiento del ordenador, el programa pondrá su ficha en la posición tres para «taponar» la jugada victoriosa del contrario.

Los lectores más atentos habrán observado que en el programa no se emplean otros criterios de valoración de los que sí hablamos en el número anterior, como por ejemplo la posibilidad de seguir colocando fichas a ambos lados del alineamiento. Lo hemos hecho así por dos razones fundamentales: en esta serie, como ya hemos advertido, los programas que proponemos no pretenden ser pautas rígidas y acabadas en todos sus detalles, sino materiales para que el lector trabaje con ellos y desarrolle sus propias mejoras y creaciones; este hecho nós obliga también a no tratar de complicar excesivamente el listado, para hacerlo accesible sin necesidad de dedicar largas horas a su estudio.

No obstante, podemos proponer una mejora que permita valorar especialmente las posiciones situadas en el centro del tablero, por ofrecer más posibilidades de completar una línea. Bastaría simplemente con sumar un punto al valor almacenado por cada

combinación si en esa posición las variables n y f son distintas de uno, de cinco y de ocho, es decir, si esa posición no está en un borde del tablero.

Entre las líneas 2000-2130, el programa trabaja todo lo relacionado con el turno del jugador. Vigila que éste no trate de colocar en un lugar en que no pueda hacerlo (si lo intenta sonará un pitido para avisarle), mueve el cursor de una posición a otra e imprime la ficha (en color verde) en el lugar elegido. Si su monitor es en blanco y negro, quizás tenga alguna dificultad para diferenciar los colores de las fichas, pero puede cambiarlos a su gusto modificando las líneas 1820 y 2110 (variando el valor de INK).

El cursor se desplaza con las teclas del cursor, y la ficha se coloca en el lugar elegido pulsando «cero». Cuando al principio del juego, el programa nos pregunta si queremos que empiece él, basta con pulsar la «s» para contestar que si, y la «n» para no. En este último caso, comenzaríamos nosotros.

A pesar de la sencillez del algoritmo empleado en este programa, observaremos que resulta difícil vencerlo, sobre todo si empieza él a jugar. Sabiendo cómo funciona y en qué se basa su análisis, podremos encontrar un sistema para ganar siempre, pero también otro sistema para que la máquina lo evite.

Finalmente, hemos de advertir que la introducción de mejoras en el programa que supongan un aumento del listado, sobre todo en la parte correspondiente al turno del ordenador, puede suponer un considerable incremento de la lentitud de la máquina en la resolución de sus jugadas.

LA RUTINA 2040

Los «devoradores» de bibliografía y revistas de informática habrán podido comprobar con cuánta frecuencia se invocan las «limitaciones del BASIC» para disculpar a lo que en algunas ocasiones no es más que el desconocimiento de las posibilidades de este lenguaje de programación. La línea 2040 del programa que acompaña a este texto, encargada de determinar qué tecla se ha pulsado y variar el valor de la posición en que se ha de imprimir el cursor, es un claro ejemplo de un correcto aprovechamiento del BASIC, en aras a una mayor velocidad y un menor gasto de memoria.

Las condicionales del tipo

IF...THEN...

son extraordinariamente lentas. Varias de ellas en sucesión colocadas en líneas distintas, pueden inutilizar un programa al hacerlo excesivamente lento. Pero como las sentencias condicionales resultan imprescindibles para casi todo lo que queremos hacer, se hace necesario idear otro tipo que consuma menos memoria y menos tiempo, como el que proponemos en la línea 2040.

Las expresiones colocadas entre paréntesis en el ejemplo que ponemos, dan siempre como resultado uno o cero, según sean verdaderas o falsas. Como los valores que tenemos que sumar o restar al valor de las variables que almacenan la posición del cursor, n y f, son siempre uno o cero, podemos utilizar las posibilidades de la sentencia AND a través de una serie de operaciones lógicas para alterar esas variables directamente.

Veamos ahora lo que hubiéramos necesitado hacer para conseguir el mismo efecto con sentencias del tipo IF ... THEN...

```
2040 IF i$=8 AND n<8 THEN LET n=n+1
2042 IF i$=5 AND n>1 THEN LET n=n-1
2044 IF i$=7 AND f<5 THEN LET f=f+1
2046 IF i$=6 AND f>1 THEN LET f=f-1
```

La comparación, como se ve, no deja lugar a dudas.

Sobre este tipo de tácticas para conseguir un mayor aprovechamiento de las posibilidades de nuestro ordenador, nos remitimos al artículo publicado en el número 6 de **INPUT**, «Exígele más a tu BASIC». No debemos olvidar que el empleo de todos los medios posibles para conseguir una mayor velocidad de proceso es imprescindible para el trabajo en materia de **Inteligencia Artificial**.

Esperamos que el lector pase un buen rato con el programa «CUATRO R.», y que no se limite a teclearlo y dejarlo tal cual está.

```
10 REM Las letras subrayadas
   deben introducirse en
   modo grafico
20 LET tt=1: DIM a(8,5)
```

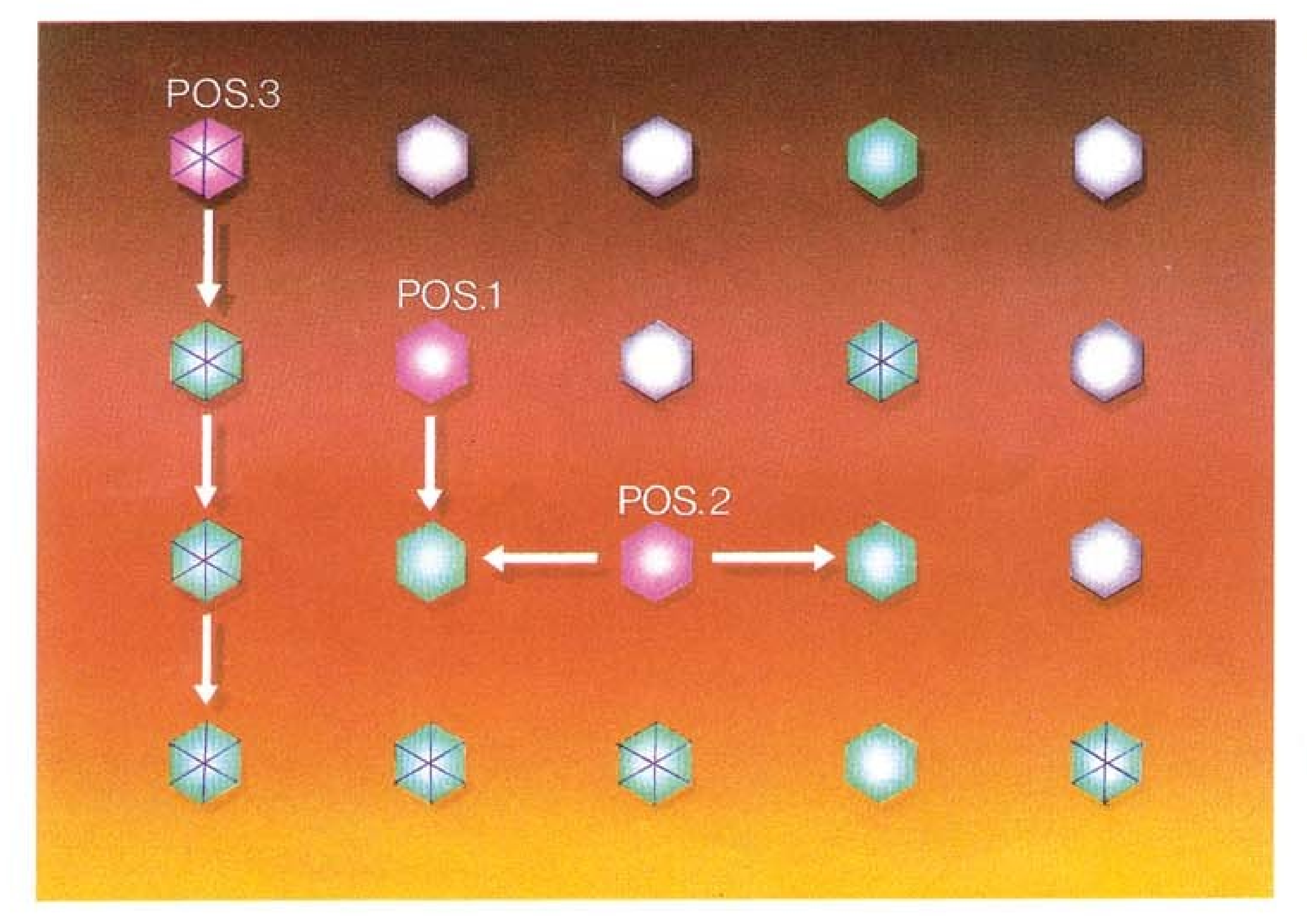


Figura 3.- Análisis de tres posiciones para ambos jugadores.




```
50 POKE 23658,0: BEEP .5,30:
   GO TO 9000
1000 REM MI TURNO
1010 IF t=2 THEN GO TO 2000
1020 LET v=0: LET c=0: PRINT
     AT 2,1;"MI TURNO[12*
     ESPACIO]": LET tf=0: LET
     tff=0
1025 IF tt=1 THEN LET nn=4:
     LET ff=1: GO SUB 1800:
     LET t=2: GO TO 2000
1028 LET uu=0: LET u=0
1030 FOR n=1 TO 8
1040 FOR f=1 TO 5
1050 IF a(n,f)=0 THEN LET uu
     =uu+1: GO SUB 1100
1060 NEXT f: NEXT n
1062 IF uu=0 THEN GO TO 2500
1065 IF tff=3 AND u=0 THEN
     LET v=1: GO TO 1800
1068 IF uu=1 THEN GO TO 2500
1070 IF u=0 THEN LET u=1: GO
     TO 1030
1075 GO SUB 1800
1080 LET tt=tt+1: LET t=2:
     GO TO 2000
1100 REM
1105 LET c=0: LET x=n: LET
     y=f
1110 LET x=x+1: GO SUB 1500:
     IF r=0 THEN GO TO 1110
1115 LET x=n: LET y=f
1120 LET x=x-1: GO SUB 1500:
     IF r=0 THEN GO TO 1120
1125 LET c=0: LET x=n: LET
     y=f
1130 LET x=x-1: LET y=y-1:
     GO SUB 1500: IF r=0
     THEN GO TO 1130
1135 LET x=n: LET y=f
1140 LET x=x+1: LET y=y+1:
     GO SUB 1500: IF r=0
     THEN GO TO 1140
1145 LET c=0: LET x=n: LET
     y=f
1150 LET x=x-1: LET y=y+1:
     GO SUB 1500: IF r=0
     THEN GO TO 1150
1155 LET x=n: LET y=f
1160 LET x=x+1: LET y=y-1:
```

Figura 4.- Estructura de la matriz a(n,f).

```
     GO SUB 1500: IF r=0
     THEN GO TO 1160
1165 LET c=0: LET x=n: LET
     y=f
1170 LET y=y-1: GO SUB 1500:
     IF r=0 THEN GO TO 1170
1175 LET x=n: LET y=f
1178 LET y=y+1: GO SUB 1500:
     IF r=0 THEN GO TO 1178
1179 LET c=0: LET x=n: LET
     y=f
1180 IF tf>tff THEN LET tff=
     tf: LET nn=n: LET ff=f
1190 LET f=5: RETURN
1500 LET g=(1 AND x<1)+(1
     AND y<1)+(1 AND x>8)+(1
     AND y>5): IF g>0 THEN
     GO TO 1520
1505 IF a(x,y)<>1 AND u=0
     THEN GO TO 1520
1508 IF a(x,y)<>2 AND u=1
     THEN GO TO 1520
```

```
1510 LET r=0: LET c=c+1:
     RETURN
1520 IF c>tf THEN LET tf=c
1530 LET r=1: RETURN
1600 REM
1800 IF v=2 THEN BEEP .1,30:
     BEEP .1,30: BEEP .1,30:
     PAUSE 20: CLS: PRINT AT
     10,9;"ME HAS GANADO";AT
     12,7;"Pulsa una tecla":
     PAUSE 0: RUN
1810 LET x=20-((ff-1)*4):
     LET y=(nn-1)*4+1
1820 PRINT INK 2; AT x,y;
     "AC"; AT x+1,y;"BD"
1830 LET a(nn,ff)=1
1840 IF v=1 THEN BEEP .5,30:
     BEEP .5,30: BEEP .5,30:
     PAUSE 100: CLS: PRINT
     AT 10,10;"TE HE GANADO";
     AT 12,8; "Pulsa una
     tecla": PAUSE 0: RUN
1850 RETURN
2000 REM SU TURNO
2005 IF t=1 THEN GO TO 1000
2010 LET n=1: LET f=5: BEEP
     .05,40: PRINT AT 2,1;
     "SU TURNO[10*ESPACIO]":
     GO TO 2060
2020 PAUSE 0: LET i$=INKEY$
2030 PRINT OVER 1;AT x,y;
     CHR$ 143
2035 IF i$="0" THEN GO TO
     2100
2040 LET n=n+(1 AND i$="8"
     AND n<8)-(1 AND i$="5"
     AND n>1): LET f=f+(1
     AND i$="7" AND f<5)-(1
     AND i$="6" AND f>1)
2050 BEEP .05,40
2060 LET x=20-((f-1)*4)-1:
     LET y=(n-1)*4+1: PRINT
     OVER 1; AT x,y; CHR$
     143: GO TO 2020
2100 IF f=1 THEN GO TO 2108
2105 IF a(n,f-1)=0 THEN BEEP
     1,20: GO TO 2060
2108 IF a(n,f)<>0 THEN BEEP
     1,20: GO TO 2060
2110 LET x=20-((f-1)*4): LET
     y=(n-1)*4+1: PRINT  INK
     4; AT x,y;"AC"; AT x+1,
     y;"BD": LET a(n,f)=2:
     LET t=1: LET tt=2: LET
     tff=0: LET tf=0: LET u
     =1: GO SUB 1100
2120 IF tff=3 THEN BEEP .1,
     30: BEEP .1,30: BEEP .1
     ,30: PAUSE 50: CLS:
     PRINT AT 10,9;"ME HAS
     GANADO"; AT 12,8;"Pulsa
     una tecla": PAUSE 0: RUN
2130 GO TO 1000
2500 REM tablas
2510 PRINT AT 10,4;"HEMOS
     QUEDADO EN TABLAS"; AT
     12,8;"Pulsa una tecla":
     PAUSE 0: RUN
9000 REM udg
9010 FOR n=USR "a" TO USR
     "a"+31
9020 READ a: POKE n,a: NEXT n
9030 DATA 3,15,63,63,127,127
     ,255,255,255,255,127,
     127,63,63,15,3,192,240,
     252,252,254,254,255,255
     ,255,255,254,254,252,
     252,240,192
9050 REM tablero
9055 INK 7: PAPER 0: BORDER
     0: CLS
9060 FOR n=4 TO 20 STEP 4
9070 PRINT AT n,0;" AC AC AC
     AC AC AC AC AC BD BD BD
     BD BD BD BD BD "
9080 NEXT N
9090 PRINT AT 0,1;"[5*
     ESPACIO]LAS CUATRO EN
     RAYA[5*ESPACIO]"
9100 PAPER 0: INK 7: PRINT
     FLASH 1; AT 2,1;"
     EMPIEZO YO ? (S/N)"
9110 LET t=(1 AND INKEY$="s"
     )+(2 AND INKEY$="n"):
     IF t=0 THEN GO TO 9100
9120 GO TO 1000
```

GANADORES DE LOS MEJORES DE INPUT SINCLAIR

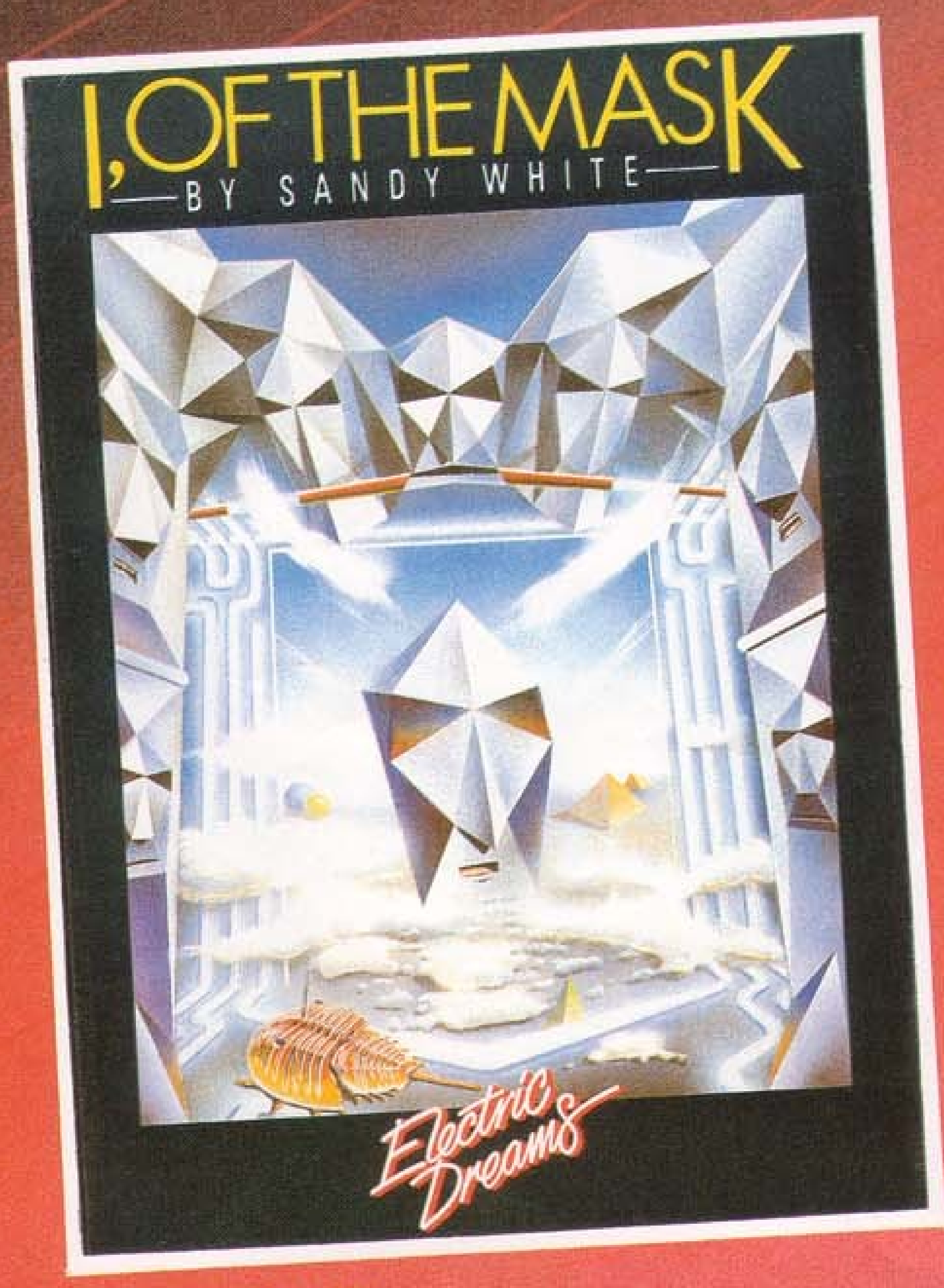
En el sorteo correspondiente al número 8 realizado entre quienes escribisteis mandando vuestros votos a LOS MEJORES DE INPUT han resultado ganadores:

NOMBRE	LOCALIDAD	JUEGO ELEGIDO
Enrique Ordaz Villareal	Cádiz	Supermán
José Antonio Gordillo López	Sevilla	Profanation
David A. Rey Caride	Pontevedra	Spy Vs Spy
J. Carlos Jódar Robles	Sevilla	Commando
Luis Remacha Montón	Badalona (Barcelona)	Commando
Francisco Cosano Cabanillas	Córdoba	Skyfox
F. Xavier Rodríguez Rodríguez	Barcelona	Sir Fred
Antonio Jesús Sánchez Marín	Cabra (Córdoba)	Supermán
Eneko Gorrichu Lozano	Bilbao	Exploding Fist



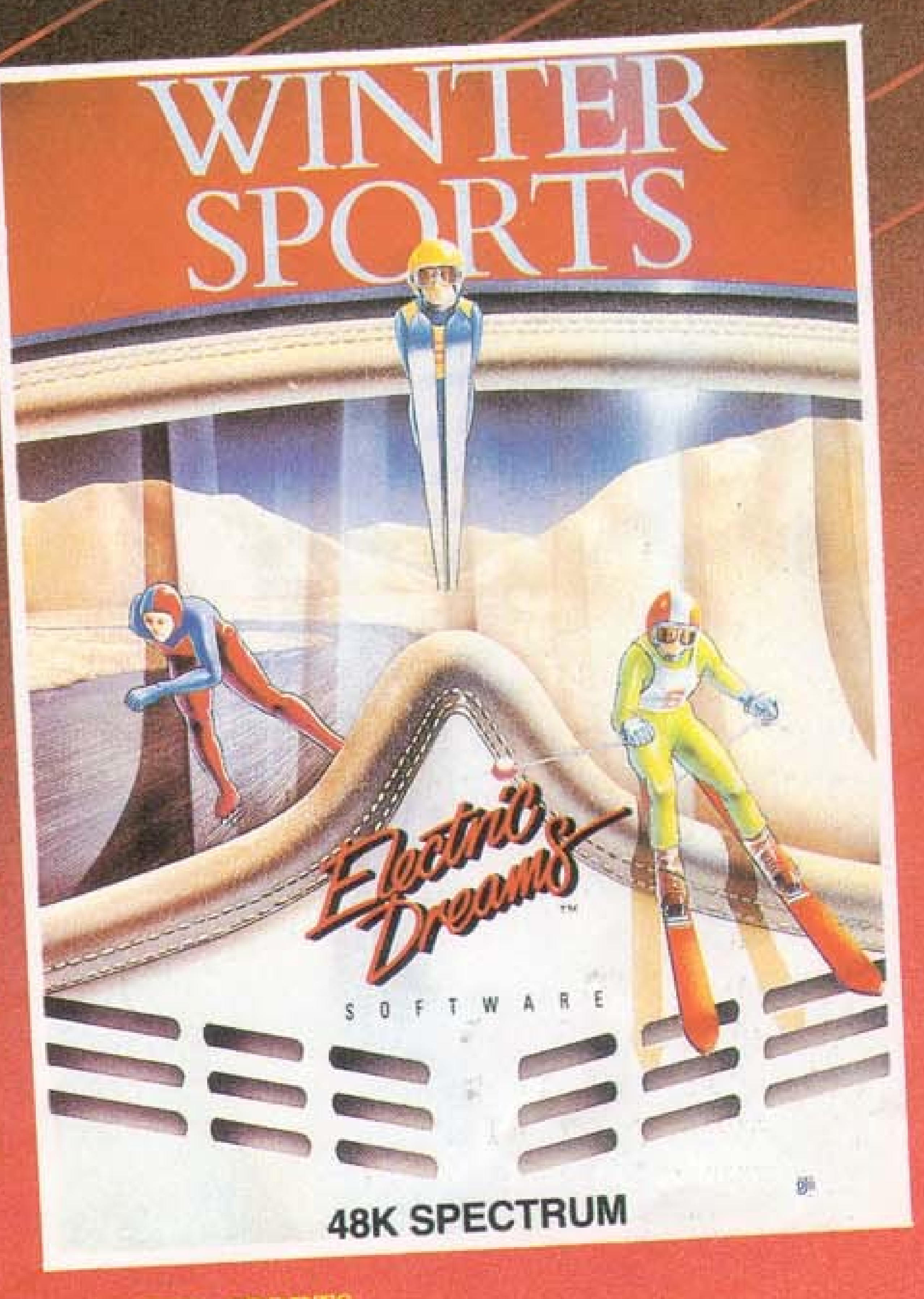
Presenta

LO MEJOR PARA TU SPECTRUM



I OF THE MASK

El laberinto mágico. Introdúcete en el movimiento tridimensional del planeta Newgama III y encuentra las piezas necesarias para completar el robot.



WINTER SPORTS

Si no puedes practicar tus deportes de invierno favoritos te ofrecemos la oportunidad de hacerlo con tu ordenador. Ocho competiciones distintas te esperan. Apresúrate a equiparte y disfruta con el descenso, slalom, salto ski, etc.

A

Disponibles para:
AMSTRAD A

EN TIENDAS ESPECIALIZADAS Y GRANDES ALMACENES, O DIRECTAMENTE POR CORREO O TELEFONO A: **PROEIN, S.A.**
Velázquez, 10 - 28001 Madrid Tels. (91) 276 22 08/09

ENTENDIENDO EL PEEK Y EL POKE

Aquí tienes la oportunidad de ejercer un control directo sobre tu ordenador, mirando lo que hay en su memoria y alterando o utilizando los valores que contiene. Y además puedes hacer todo esto también en BASIC.

Casi siempre puedes utilizar tu ordenador sin preocuparte para nada de la forma en que trabaja su memoria. Por ejemplo, cuando escribes LET A = 67, el ordenador ya se ocupa de disponer posiciones libres de la memoria a las cuales asigna la etiqueta «A» y almacena en ellas el valor 67. Cuando más tarde tú tecleas PRINT A, el ordenador sabe exactamente adónde tiene que ir para encontrar de nuevo a A. Todo esto se hace automáticamente y solamente cuando empiezas a aprender el lenguaje máquina se hace necesario decir al ordenador qué direcciones de memoria tiene que usar.

Pero existe una manera de observar desde el BASIC la memoria del ordenador, que te permite utilizar realmente en tus programas los valores almacenados en ella. También puedes cargar tus propios valores para modificar la forma en que se comporta el ordenador.

Las herramientas de que se sirve el BASIC para hacer esto son PEEK y POKE. El PEEK te permite observar un valor almacenado en la memoria, mientras que el POKE se utiliza para cargar en memoria tus propios valores.

COMO FUNCIONAN EL PEEK Y EL POKE

Tu ordenador **Spectrum** de 48K es capaz de direccionar 65536 posiciones de memoria, que es a lo que se llama 64K. Los ordenadores de 16K tienen 32768 direcciones de memoria. Una parte de esta memoria es ROM, que es la memoria en la que sólo se puede leer pero no escribir. El contenido de la ROM es fijo por lo que, si bien puedes leer lo que hay en ella utilizando el PEEK, no puedes POKEar nada en ella para modificar su contenido. La

PEEK 12460

77 29 90 87 58 45 56 189 6 192 255 1 98 37 2 3 94 76 80 59 232 60 34 80 55 12460 2461

- EXAMINA LA MEMORIA
- COMO UTILIZAR PEEK Y POKE
- POKEANDO CARACTERES EN LA PANTALLA
- LECTURA DEL RELOJ

parte restante de la memoria es RAM, que significa memoria de acceso aleatorio, también llamada memoria de lectura y escritura. En esta parte de la memoria puedes hacer tanto PEEK como POKE, siendo aquí donde se almacenan los programas y las variables del BASIC.

El siguiente programa te permitirá observar cualquier posición de la memoria de tu ordenador, tanto en la ROM como en la RAM:

Teclea para Spectrum y ZX81

```
10 INPUT "DIRECCION...";D
20 N=PEEK D
30 PRINT "CONTENIDO....";N
40 PRINT
50 GOTO 10
```

Teclea el número que quieras entre 0 y 65535 y podrás ver lo que hay en esa posición de memoria, aunque si haces un PEEK en determinadas zonas puede que obtengas un valor falso en lugar del valor que realmente hay allí almacenado.

El contenido de la RAM dependerá de lo que estés haciendo en ese momento con tu ordenador, pero el contenido de la ROM es fijo.

Observa que siempre obtienes números enteros entre 0 y 255. En hexadecimal son números entre 0 y FF, lo que significa que son bytes (un byte es un número hexadecimal de dos dígitos). Toda posición de memoria contiene un solo byte y cualquier número mayor no puede ser almacenado en una única posición de memoria. Si le sumas 1 al número hexadecimal FF, obtendrás el número 0100. Este número requiere ya dos bytes para su representación por lo que para ser almacenado necesita dos posiciones de memoria: 01 en una de ellas y 00 en la otra.

Con el programa siguiente puedes POKEar números en la memoria. Por el momento POKEa solamente los bytes de uno en uno, es decir valores entre 0 y 255.

Teclea para Spectrum y ZX81

```
10 PRINT"CONTENIDO.......";
   PEEK 30000
20 INPUT "NUMERO(0-255)..";N
30 POKE 30000,N
40 PRINT"NUEVO CONTENIDO.";
   PEEK 30000
45 PRINT
50 GOTO 20
```

En primer lugar el programa presenta el contenido de la posición de memoria y a continuación POKEa tu número en la misma y presenta su contenido por segunda vez para que puedas comprobar que tu número está realmente allí.

Puedes cambiar desde esa dirección de memoria a cualquier otra dirección que desees. Observa que si intentas POKEar algo en ROM no sucede nada, aunque no pierdes nada por intentarlo. Puede que tu sistema se cuelgue o haga cosas extrañas si POKEas algo en determinadas zonas de la RAM, pero tampoco en este caso sucederá nada grave. Limítate a apagar unos momentos tu máquina para inicializar la memoria.

Intenta ahora POKEar un número mayor que 255 y observa lo que sucede. Como sólo puedes poner un byte en cada posición, recibirás un mensaje de error.

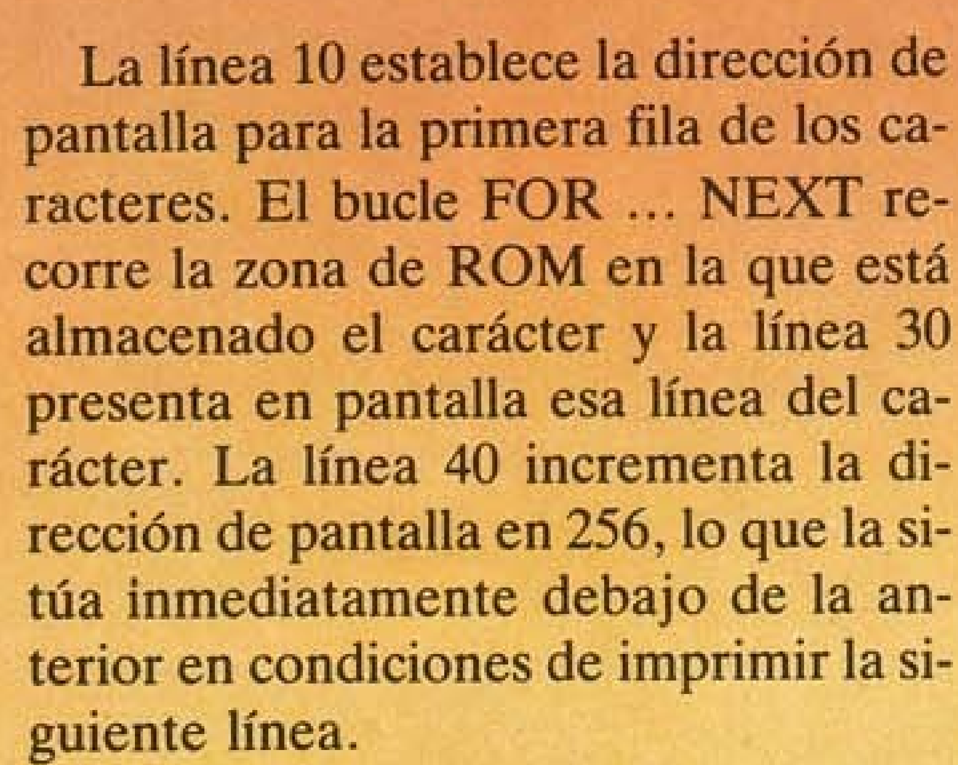
POKEANDO EN PANTALLA

Si observas los mapas de memoria que vienen en el manual, verás que hay una sección de la memoria dedicada íntegramente a la presentación en pantalla. POKEando determinados números en esa zona de memoria, aparecerán realmente caracteres en la misma. Un carácter, igual que un GDU, está constituído por un conjunto de puntos en una retícula de 8×8. Cada línea ha de ser POKEada separadamente en la pantalla para construir todo el carácter.

Afortunadamente las formas de los caracteres están ya almacenadas en ROM por lo que puedes hacer un PEEK en ROM para mirar cada línea y POKEarla después en la pantalla. El siguiente programa POKEa una A en la parte superior izquierda de la pantalla:

Teclea para Spectrum

```
10 LET dest=16384
20 FOR a=15880 TO 15887
30 POKE dest,PEEK a
40 LET dest=dest+256
50 NEXT a
```



La línea 10 establece la dirección de pantalla para la primera fila de los caracteres. El bucle FOR ... NEXT recorre la zona de ROM en la que está almacenado el carácter y la línea 30 presenta en pantalla esa línea del carácter. La línea 40 incrementa la dirección de pantalla en 256, lo que la sitúa inmediatamente debajo de la anterior en condiciones de imprimir la siguiente línea.

Aunque este método funciona, resulta algo lento, por lo que casi siempre es mejor utilizar PRINT AT o PRINT TAB.

PEEKs EN ROM Y EN RAM

Al ejecutar el primer programa para observar la memoria del ordenador, todo lo que obtendrás serán números entre 0 y 255. Sin embargo muchos de estos números son realmente códigos ASCII de letras, muchas de las cuales torman parte de palabras o frases.

Si no estás seguro de los códigos ASCII de que se trata, puedes consultar la tabla que acompaña al artículo.

El programa siguiente lee toda la ROM de tu ordenador y convierte los números en caracteres antes de visualizarlos en pantalla:

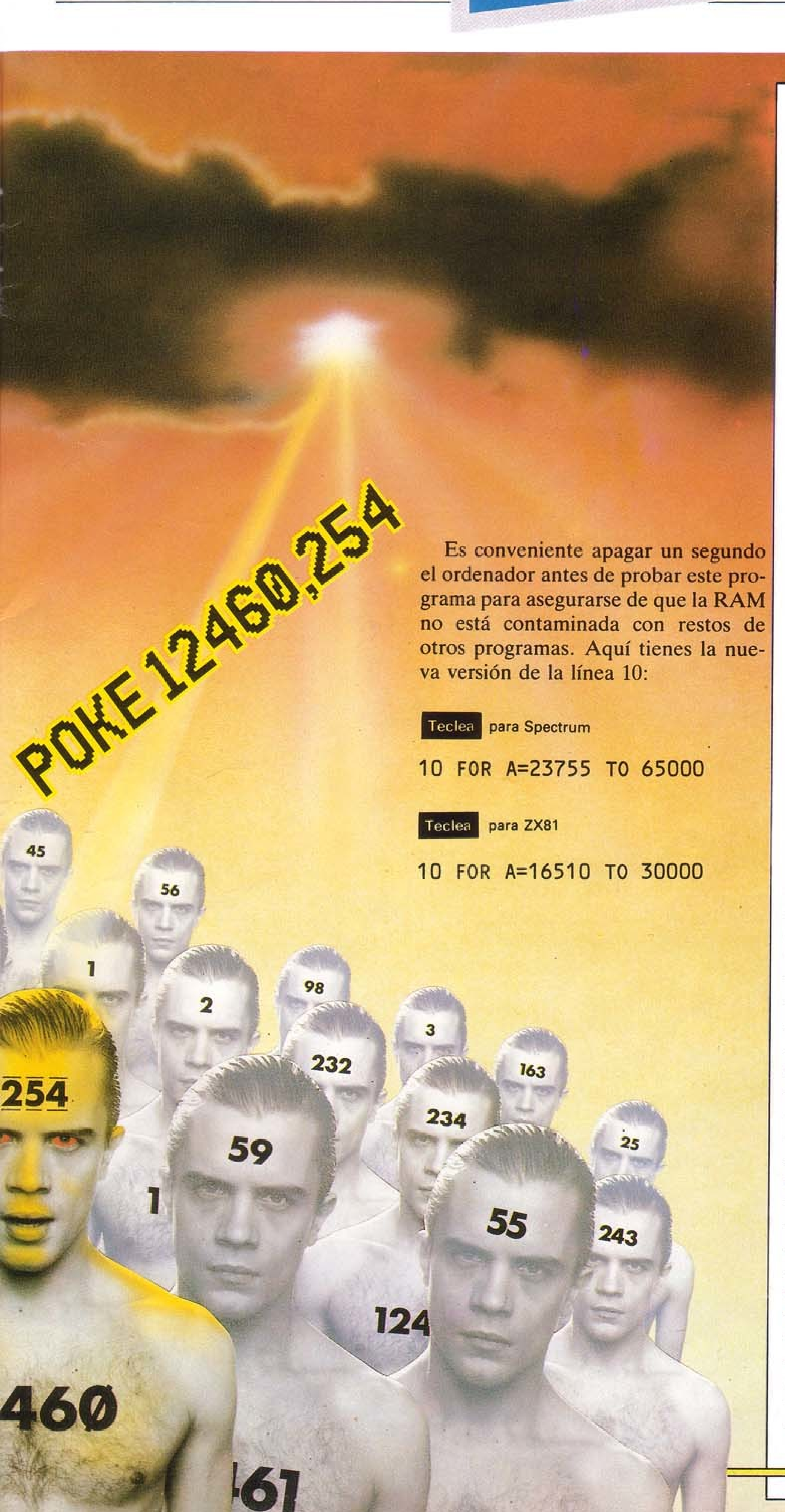
Teclea para Spectrum y ZX81

```
10 FOR A=0 TO 16383
20 LET N=PEEK A
30 IF N>31 AND N<127 THEN
   PRINT CHR$ N;
40 NEXT A
```

Las líneas 10 y 40 hacen que se recorra toda la memoria del ordenador, la línea 20 realiza un PEEK en cada posición de memoria y la 30 transforma el número en un carácter y lo imprime. La línea 30 limita también el campo de números sobre los que se efectúa la conversión para que el ordenador no intente imprimir códigos de control o símbolos gráficos.

Puedes también imprimir el contenido de la RAM exactamente de la misma forma; para ello no tienes más que modificar las direcciones de memoria que figuran en la línea 10. Las direcciones que se dan más adelante imprimen una parte de la zona donde se almacenan los programas en BASIC. Esto te permite contemplar el programa tal como realmente está almacenado en la memoria.





Es conveniente apagar un segundo el ordenador antes de probar este programa para asegurarse de que la RAM no está contaminada con restos de otros programas. Aquí tienes la nueva versión de la línea 10:

Teclea para Spectrum

```
10 FOR A=23755 TO 65000
```

Teclea para ZX81

```
10 FOR A=16510 TO 30000
```

TABLA DE CODIGOS ASCII

Cada uno de los caracteres que utiliza tu ordenador tiene un número de código y la mayoría de los ordenadores emplean un código estándar llamado código ASCII. El significado de las siglas es Código Americano Normalizado para Intercambio de Información (American Standard Code for Information Interchange).

Estos códigos son los números usados con CHR$ y CODE. CHR$ transforma el número en un carácter, mientras que CODE realiza la conversión inversa, transformando un caracter en su número de código.

Además, cada vez que el ordenador almacena una palabra en su memoria, lo que se almacenan son los códigos ASCII de los correspondientes caracteres.

Tabla ASCII

Código número	Carácter ASCII	Código número	Cárácter ASCIII
32	espacio	62	>
33	!	63	?
34	"	64	@
35	#	65	A
36	$	66	B
37	%	67	C
38	&	68	D
39	'	69	E
40	(	70	F
41	)	71	G
42	★	72	H
43	+	73	I
44	,	74	J
45	–	75	K
46	.	76	L
47	/	77	M
48	0	78	N
49	1	79	O
50	2	80	P
51	3	81	Q
52	4	82	R
53	5	83	S
54	6	84	T
55	7	85	U
56	8	86	V
57	9	87	W
58	:	88	X
59	;	89	Y
60	<	90	Z
61	=		



USO DEL PEEK Y EL POKE

Lo que hemos hecho hasta ahora ha sido observar en general lo que hay en la memoria del ordenador y POKEar caracteres en la pantalla. Esto puede darte una buena idea de la forma en que funcionan el PEEK y el POKE pero no es en sí mismo especialmente útil.

Para hacer algo verdaderamente útil tienes que examinar direcciones de memoria específicas. Por ejemplo la dirección de memoria 23609 controla en el **Spectrum** el sonido que se genera desde el teclado al pulsar una tecla. Normalmente contiene el valor 0, que corresponde a un breve «click», pero si pones un número mayor (puedes llegar hasta 255) se obtiene un pitido más largo. Con la instrucción POKE 23609,80 se obtiene un razonable chasquido.

En el **Spectrum** las instrucciones PEEK y POKE suelen usarse sólo ocasionalmente, con frecuencia para modificar la forma en que trabaja el teclado o para alterar la presentación en pantalla.

Aquí tienes alguna de las cosas que puedes ensayar en tu ordenador.

Ya hemos visto la manera de utilizar el POKE para alterar el sonido del teclado. El siguiente ejemplo permite modificar el intervalo de tiempo que transcurre desde que se pulsa una tecla hasta que ésta empieza a hacer la autorrepetición; esto resulta muy útil para dar velocidad a juegos que dependen de pulsaciones de teclas para mover caracteres. En el ejemplo que sigue a continuación y en los siguientes, X es una variable a la que debes asignarle un valor.

Teclea para Spectrum y ZX81

```
POKE 23561,X
```

Normalmente a X se le asigna el valor 35 cuando se enciende el ordenador, por lo que deber utilizar un valor inferior a 35 si quieres disminuir el retardo o un número mayor (puedes llegar hasta 255) para retardos mayores.

También es posible modificar la propia autorrepetición tecleando:

```
POKE 23562,X
```

Esta vez el valor normalmente asignado a X es 5. Utiliza pues X = 1 para tener una autorrepetición rápida y X = 255 para tener una muy lenta.

MEDIDA DE TIEMPOS

El **Spectrum** no tiene una instrucción TIME, por lo que para usar el reloj interno del ordenador tienes que recurrir a hacer PEEKs en memoria. La hora se almacena en tres bytes situados en tres posiciones consecutivas de la memoria.

```
PRINT(PEEK 23672+256*PEEK
23673+65536*PEEK 23674)
```

La sentencia anterior te dirá cuántos cincuentaavos de segundo ha estado encendido tu **Spectrum** desde que tecleaste la última instrucción NEW. Para poner a cero el reloj, pulsa NEW o haz POKE 0 en las tres direcciones de memoria.

OTROS POKEs

Aquí te presentamos otros dos POKEs más que te resultarán muy útiles cuando estés escribiendo programas que han de ser utilizados por otras personas. El primero te permite asegurarte de que todas las letras tecleadas aparecerán como mayúsculas:

```
POKE 23658,8
```

Esto te será muy útil cuando quieras que todas las entradas sean consistentes. Para volver al modo normal, teclea POKE 23658,0.

El segundo POKE modifica el cursor, cambiándolo por cualquiera de los caracteres del teclado e incluso por palabras completas, dependiendo del número que pongas en el POKE:

```
10 FOR X=1 TO 255
20 POKE 23617,X
30 INPUT a$
40 NEXT X
```

La línea 30 contiene una sentencia INPUT para hacer que aparezca un cursor en la pantalla. En cuanto le proporciones una respuesta, el programa presentará el siguiente cursor. Los más interesantes se encuentran situados en torno a X = 200 hasta 230. Por ejemplo con X = 210 tendrás un cursor que dice CONTINUE, mientras que con X = 225 obtendrás un signo ?.





LAS TRES LUCES DE GLAURUNG

UN PROGRAMA HECHO EN ESPAÑA QUE ESTA SORPRENDIENDO EN EUROPA

Un guerrero va a enfrentarse, sólo, a los incontables peligros que acechan en el Castillo bajo la Montaña, más allá de donde alcanza la luz del Sol y de donde se atreven a llegar los corazones más valerosos.

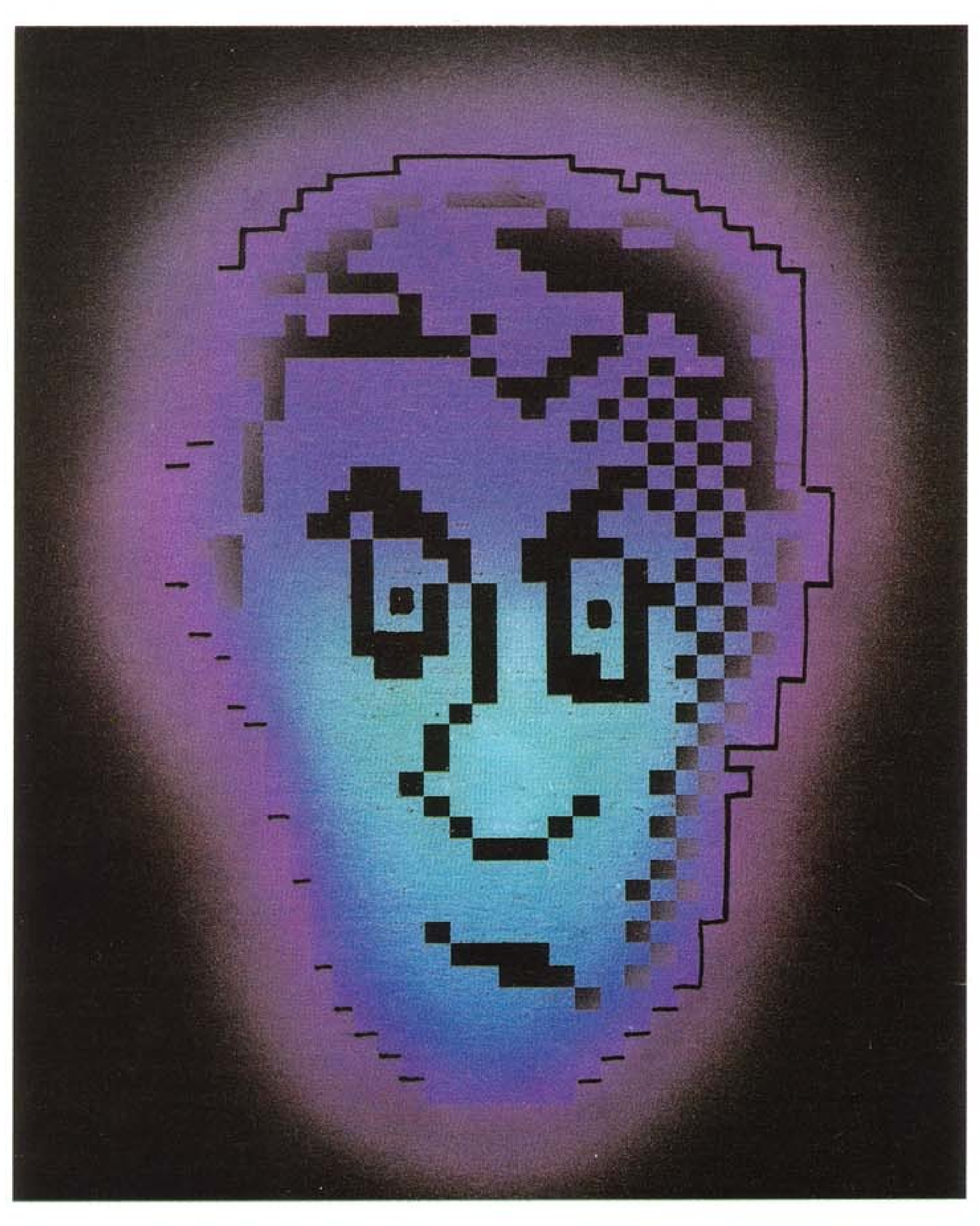
La fuerza, la astucia y la habilidad, van a medirse con el hierro, el fuego y la hechicería, en uno de los más tremendos choques entre el Bien y el Mal que el Universo ha presenciado jamás.

ERBE

DISTRIBUIDOR EXCLUSIVO PARA ESPAÑA
ERBE SOFTWARE.
SANTA ENGRACIA, 17. Tel: 447 34 10
DELEGACION BARCELONA.
Avd. MISTRAL, 10. Tel. (93) 432 07 31

NO TE CONFORMES CON 21 UDG, EXIGELE MAS A TU SPECTRUM

Podemos estar seguros de que muchos de nuestros lectores se habrán preguntado en más de una ocasión (el autor de este artículo lo hizo frecuentemente en sus tiempos de «aprendiz»), cuál es el secreto para conseguir la superabundancia de gráficos en algunos programas comerciales.



Es admirable cómo los grandes programadores se superan programa a programa burlándose del límite de los 21 UDG disponibles en el **Spectrum**, utilizando en sus creaciones cientos de espectaculares gráficos que nos sorprenden tanto por su número y variedad como por su diseño.

Pues bien, aquí no os vamos a desvelar todos sus secretos, entre otras razones porque no los compartimos, pero si os vamos a mostrar la forma de conseguir multiplicar el número de gráficos de vuestros programas a capricho. No os asustéis: NO se trata del tan manido sistema de «bloques», que tantas veces se ha presentado como la solución mágica y tantas veces nos ha desilusionado, sino de un recurso avanzado y verdaderamente «profesional».

El sistema de bloques ha sido bastante utilizado hasta el momento por los aficionados a la programación como único medio de salvar el escollo de los 21 UDG en programas que necesitan un número mayor de gráficos. Consiste simplemente en almacenar en memoria varios grupos de gráficos en direcciones distintas y variar el puntero de UDG según el grupo de gráficos a que se quiera acceder.

Esto tiene el grave inconveniente de que no se pueden usar simultáneamente más de 21 gráficos distintos, ya que al usar un bloque se excluyen todos los demás.

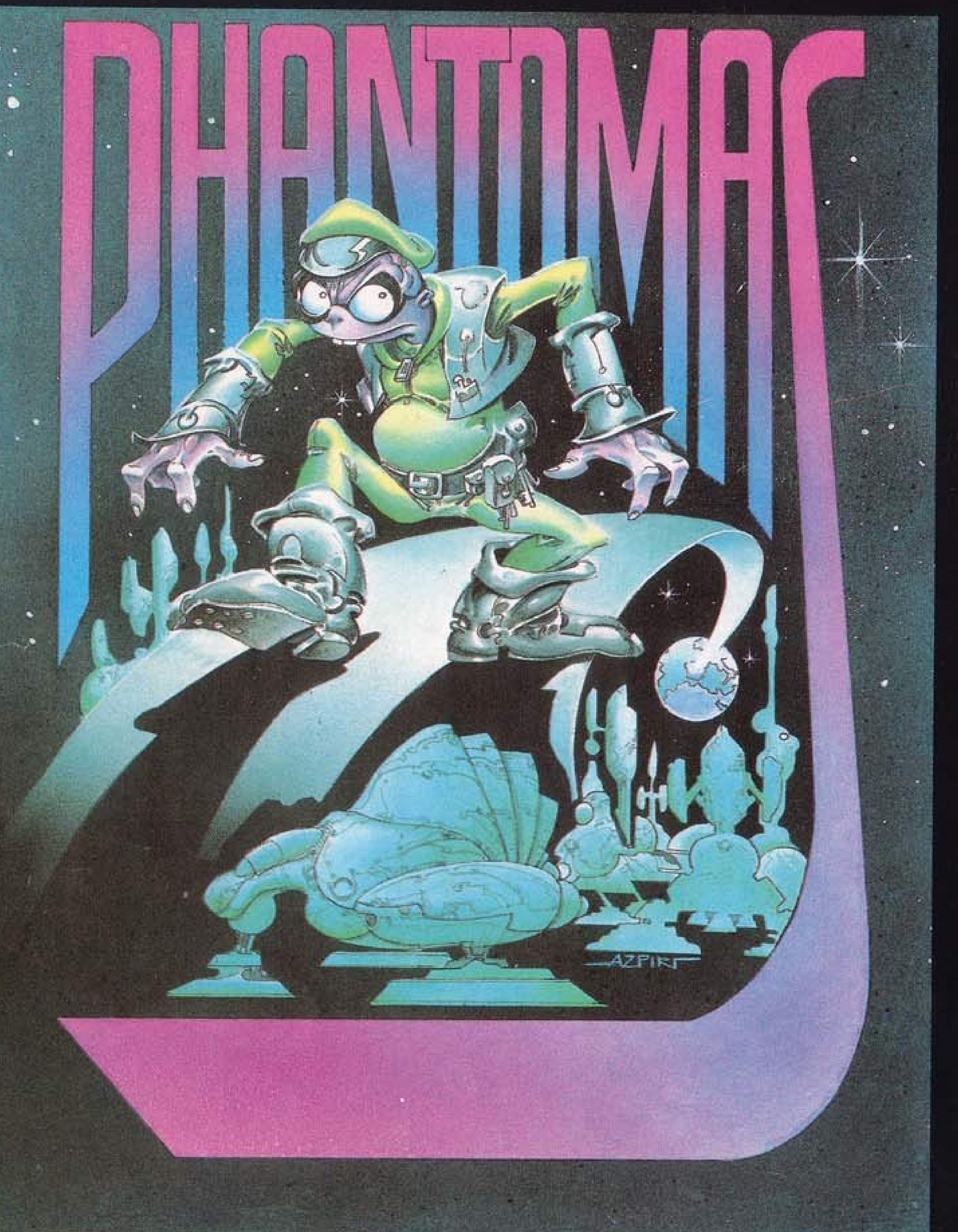
Otro problema generado por esta limitación gráfica del **Spectrum**, es la necesidad de depender de su juego de caracteres estandar, cuando muchas veces se desea poder utilizar otro juego más acorde con el tema del programa, con letras más grandes, más pequeñas, futuristas, góticas, inclinadas, o simplemente con símbolos en castellano.

La solución definitiva que aquí vamos a proponer, supone la posibilidad de disponer de un repertorio gráfico de más de cien caracteres diseñados a vuestro gusto. ¿Hay quién de más?

LA VARIABLE DEL SISTEMA «CHARS» Y EL JUEGO DE CARACTERES

A partir de la posición de memoria 15616, se encuentran almacenados en la ROM del **Spectrum** los 768 bytes correspondientes a los 96 gráficos del juego de caracteres, numerados con códigos ASCII del 32 al 127. Si tomamos el manual veremos que dicho jue-

Tuya es la elección





Plaza España, 18
Torre de Madrid, 29-1.
28008 MADRID

go de caracteres incluye todos los símbolos desde el espacio en blanco (con el código 32), hasta el «copy» (con el 127), a los que podemos acceder desde el teclado, con excepción de aquellos que son compuestos (como PEEK o PRINT).

Existe una variable del sistema que indica al ordenador a qué dirección debe dirigirse para encontrar el juego de caracteres. Se trata de la variable «CHARS», situada en las posiciones 23606 y 23607. Si cambiamos el contenido de esta variable, podemos hacer que el ordenador se dirija a un nuevo juego de caracteres almacenado por nosotros en otro lugar de la memoria cada vez que le ordenemos imprimir un caracter. De esta forma, dispondremos de 96 gráficos creados a nuestro gusto a parte de los 21 U.D.G., a los que podremos acceder utilizando los códigos ASCII del juego de caracteres estandar. Esto supone la posibilidad de que el ordenador imprima letras de formato distinto a las normales, y que las letras y símbolos que no necesitemos (por ejemplo las minúsculas, los signos matemáticos y los paréntesis y corchetes) sean sustituidos por otros gráficos que nos sean más útiles.

Si inspeccionamos el contenido de la variable CHARS, veremos que no almacena el número de comienzo del juego de caracteres, sino ese número (15616) menos 256, es decir (15360). Esto es así porque cuando pedimos al ordenador que imprima un caracter, lo busca multiplicando su código por ocho y sumándole después el contenido de la variable CHARS. Veamos un ejemplo:

Si queremos imprimir la letra «a», que tiene el código 97, tendremos:

$$97 * 8 = 776$$
$$15360 + 776 = 16136$$

Y si queremos imprimir un espacio en blanco (con el código 32):

$$32 * 8 = 256$$
$$15360 + 256 = 15616$$

es decir, la dirección de comienzo del juego de caracteres.

La variable CHARS almacena siempre números mayores de 255, y por tanto ocupa dos bytes. El menos significativo se almacena en la dirección 23606, y el más significativo en la dirección 23607. Para redefinir todo el juego de caracteres con 96 nuevos gráficos, a parte de los 21 de la U.D.G., debemos dejar un espacio libre en la memoria de 768 bytes. A fin de evitar el peligro de que un programa en BASIC demasiado largo se nos «escriba» sobre el nuevo banco de caracteres, elegimos una posición alta de memoria para colocar la nueva dirección de comienzo (lo que suele hacerse para las rutinas código-máquina). Aquí proponemos los 768 bytes inmediatamente anteriores a la U.D.G. como lugar más adecuado, aunque teóricamente sirve cualquier otro.

La dirección de comienzo de la U.D.G. es 65368 (en un **48K**), por tanto, la dirección a partir de la cual almacenaremos el nuevo juego de caracteres será:

$$65368 - 768 = 64600$$

Antes de comenzar a almacenar los

gráficos, hemos de bajar la RAMTOP a 64599 (para poder utilizar esta zona de la memoria desde la dirección 64600 inclusive), haciendo CLEAR 64599 (CLEAR 31831 en un **16K**).

REDEFINIENDO EL JUEGO DE CARACTERES

Para definir nada menos que 96 gráficos no podemos pedir a nadie (por mucha vocación que se tenga), que los diseñe uno por uno en un papel, y los teclee byte por byte hasta totalizar 768 cifras. Por esta razón acompañamos el artículo con un pequeño programita que, con ayuda del generador de gráficos que publicamos en los números uno y tres de INPUT (o cualquier otro generador de gráficos que tengáis), hará por vosotros la parte más tediosa del trabajo. Su funcionamiento es el siguiente:

Primero se crean varios bandos de UDG (hasta totalizar 96 gráficos) y se almacenan sucesivamente en cinta, con ayuda del generador de gráficos. Después se teclea el programa que a continuación listamos (o se carga si ya lo teníamos grabado) y se pone en marcha haciendo RUN. Nos pedirá que carguemos el primer banco de gráficos, lo colocará en la parte de memoria que le corresponda, y nos pedirá que hagamos lo mismo con el siguiente, hasta terminar.

Una vez ubicados los 96 gráficos, debemos cambiar el contenido de la variable CHARS para poder acceder al nuevo juego de caracteres. Esto lo haremos tecleando:

POKE 23606,88: POKE 23607,251

A partir de este momento, cada vez que pulsemos una tecla, se imprimirá no el caracter original, sino uno de los gráficos por nosotros definidos. Utilizando el apéndice del manual (en el que se lista el juego de caracteres completo y sus códigos), podremos planificar el nuevo juego de caracteres de forma que los gráficos que diseñemos coincidan con la tecla adecuada. Por ejemplo, si queremos crear un nuevo juego de letras con una tipografía distinta, debemos tener en cuenta que cuando pulsemos la «a», se imprimirá en pantalla el gráfico que haga el número 66 de los diseñados, ya que la «a» tiene el código 97 y el primer caracter del juego de caracteres tiene el 32 (97 − 32 = 65 ; 65 + 1 = 66).



Teclea

```
20 CLEAR 64599: LET c=64600
30 FOR n=1 TO 5
40 CLS : PRINT AT 21,0;
   "CARGA BLOQUE UDG ";n
50 LOAD ""CODE
60 FOR f=USR "a" TO USR "u"
   +7
70 POKE c,PEEK f: LET c=c+1
80 NEXT f
90 NEXT n
```

Ahora, ya podéis realizar varios modelos de letras que pueden hacer más atractivos vuestros juegos. Otra aplicación muy interesante de la redefinición del juego de caracteres es la creación de vocales con acento, la letra «ñ», etc.

RECETARIO PRACTICO DE CODIGO MAQUINA PARA EL SPECTRUM

¿Cómo se manejan los UDGs para conseguir mayor rapidez en tus juegos? ¿Cómo ganar en velocidad y realismo? Las pautas a seguir para lograr esto y muchas más cosas, serán desveladas en este recetario mensual, que interesará incluso a los no iniciados.

No vamos a ofrecer un curso completo de código máquina que, por otro lado, podría llegar a ser tedioso. Para ello hay buenos libros de aprendizaje. Por el contrario, afrontaremos la aplicación del código máquina de una forma que pretende ser amena y práctica. Desde estas páginas irán saltando rutinas, tablas y otras utilidades que puedan ir ampliando vuestro taller de ***software***.

Un programa en c.m. no es ni más ni menos, que el conjunto de instrucciones que debe realizar el microprocesador para tratar la información. Y para ello, se ayuda de los denominados registros del **Z80**, al igual que para manejar una calculadora no programable nos ayudamos de sus memorias, o del papel y el lápiz, para anotar los resultados intermedios.

En la memoria conviven tanto los datos como las instrucciones que debe realizar el microprocesador.

Los contenidos en la ROM permanecen invariables indefinidamente y los que se alojan en la RAM son los que podemos variar a nuestro antojo.

El sistema operativo del **Spectrum** no es ni más ni menos que un programa realizado en c.m. situado en la ROM, que se inicializa nada más conectarlo a la red, y que permite al ordenador comunicarse con el exterior.

En anteriores artículos hemos dejado claro donde debemos colocar el c.m. como utilizar un ensamblador o en su defecto como ensamblar un programa manualmente...

Con esta pequeña introducción estamos en condiciones de empezar.

MANEJO DE CARACTERES EN PANTALLA

Antes de empezar a escribir, aunque parezca una tontería, tenemos que saber donde queremos escribir. Lo mismo ocurre con el **Spectrum**, para escribir una cosa tiene que saber donde escribirla.

Desde el c.m. tenemos que indicar al microprocesador por que camino (canal) queremos que envíe la información al exterior. Para escribir en pantalla podemos elegir dos caminos diferentes, uno hacia la parte alta (las 22 líneas que normalmente utilizamos) y el otro hacia la parte baja (las que utiliza el ordenador para editar una línea BASIC, escribir los mensajes de error, etc. ...).

La forma de indicar el camino es la siguiente: Primero se carga el acumulador (registro A) con el valor 2 (si deseamos escribir en la zona alta) o el 0 (si queremos hacerlo en la baja) y después llamamos a la subrutina CHAN-OPEN, la encargada de acondicionarnos los caminos.

Una vez indicado el camino, imprimir un caracter es muy sencillo. Cargamos el acumulador con el valor del caracter que queremos imprimir y hacemos un RST #10 (instrucción equivalente a CALL #0010, pero más rápida y ocupa un byte menos).

El proceso anterior implicará la utilización de la rutina de impresión de caracteres de la ROM. No debemos olvidar que trabajar en c.m. es ya de por sí complicado, y debemos tratar de utilizar bien todos los recursos. La ROM es una maravillosa caja de rutinas que debemos saber manejar a la perfección, ya que sería inútil tratar de crear de nuevo cada rutina a utilizar, basta el «CALL» adecuado a cada caso.

La rutina RST #10 admite la impresión de caracteres con distintos colores de tinta, papel, *flash*, etc... utilizando los caracteres de control adecuados, siendo los más utilizados y su codificación ASCII. (Ver tabla 1.)



La utilización de dichos caracteres de control en c.m. sería muy parecida a su equivalencia en BASIC. Pruebe por ejemplo:

```
10 PRINT CHR$ 16+
   CHR$ 6+CHR$ 17+
   CHR$ 0+CHR$ 22+
   CHR$ 10 +CHR$
   7+CHR$ 18+CHR$
   1+ " INPUT
   SINCLAIR "
```

TABLA 1

```
INK -----> 16 (#10)
FLASH ---> 18 (#12)
INVERSE -> 20 (#14)
PAPER ---> 17 (#11)
BRIGHT --> 19 (#13)
AT ------> 22 (#16)
```

equivalente a:

```
10 PRINT INK 6;PAPER 0;AT 10
   ,7;FLASH 1;" INPUT
   SINCLAIR "
```

A continuación presentamos una serie de rutinas en c.m. con su correspondiente cargador en BASIC para aquellos que carezcan de un programa ensamblador. Sólo puntualizar que para la confección de las rutinas se ha utilizado el **GENS3 (DVPAC)**, con ligeros matices sin importancia, como el representar los números en hexadecimal con el símbolo « # ».

RUTINA NUMERO 1

Como ejemplo de todo lo explicado anteriormente presentamos la siguiente rutina consistente en el rellenado de la pantalla con el caracter «#» (CHR$ 35), impreso en tinta azul. Los comentarios de la rutina van separados por «;» a modo de «REMs». (Ver Rutina 1.)

El proceso se realiza 704 veces por el bucle introducido. El BC es un registro doble y B almacena el byte alto. Al incrementar el BC (línea 130), cuando se llega a 256 el registro B aumenta en una unidad y el C se pone a 0. Si ocurre tres veces, al compararlo con A (líneas 140, 150), se llega a A=B y volveremos al BASIC (RET). Mientras no ocurra esto y el *flag* Z o indicador de cero no se active, estaremos saltando a la instrucción indicada por la etiqueta BUCLE. Así haremos el proceso 256*3−64=704 veces. (Partimos con 64 en el registro contador, línea 60).

El cargador BASIC será:

```
10 CLEAR 59999:LET N=60000:
   FOR H=0 TO 26:READ A:POKE
   N+H,A:NEXT H
20 DATA 205,107,13,62,2,205,
   1,22,1,64,0,62,16,215,62,
   1,215,62,35,215,3,62,3,
   184,32,241,201
```

RUTINA 1

```
10          ORG  60000       ;Posicion de ubicacion de la rutina
20          ENT  60000       ;Direccion de ejecucion (con GENS3)
30          CALL #0D6B       ;Llamada a la rutina de borrado
40          LD   A,2         ;Introduce el valor 2 en acumulador
50          CALL #1601       ;Abre canal de pantalla
60          LD   BC,64       ;Carga el registro para iniciar bucle
70 BUCLE    LD   A,#10       ;Indica al ordenador que va a escoger
80          RST  #10         ;la tinta
90          LD   A,1         ;Tinta azul
100         RST  #10         ;Pone en memoria la opcion elegida
110         LD   A,35        ;Codigo ASCII del caracter "#" a
120         RST  #10         ;visualizar
130         INC  BC          ;Incrementa el contenido de BC en uno
140         LD   A,3         ;Carga el acumulador con 3
150         CP   B           ;Compara el valor de A con el de B
160         JR   NZ,BUCLE    ;Si A<>B salta a BUCLE (linea 70)
170         RET              ;Indispensable para volver al BASIC
```

Para ejecutar la rutina RANDOMIZE USR 60000

Observando ambos listados podemos hacer que imprima otro caracter, cambiando el código ASCII en la línea 110, o modificando directamente el caracter decimonoveno en la lista de DATAs (el 35).

Para comparar la rapidez del c.m. con la sencillez del BASIC podemos rodar el programa equivalente:

```
100 FOR N=1 TO 704: PRINT
    INK 1;"#";:NEXT N
```

RUTINA NUMERO 2

Semejante a la anterior pero introduciendo importantes variantes:

- Hace uso de la variable del sistema 23693 ATTR P para indicar de una manera directa y en una sola instrucción, la tinta, el papel, el brillo y el *flash*. En nuestro caso hemos introducido el número 79, porque pasado a binario queda de la forma que se indica en la tabla 2.

De esta manera es como se guarda en la memoria de atributos. Proponemos a nuestros lectores que prueben con distintos valores según lo expuesto anteriormente.

- Utiliza la rutina de coloración del borde 8859 (mediante CALL, línea 60), tomando el valor que introducimos previamente en A (línea 50).

- Ejecuta otra rutina distinta de borrado (#0E44), que limpia la pantalla desde la línea 24 hacia arriba, borrando el número de líneas especificado en B (línea 90).

- El bucle que utiliza se repite 32 veces. Compara A=32 con B (línea 220) que va incrementándose desde 0 (línea 110), y hace un RETorno al BASIC si A=B. Hasta que no ocurra lo anterior saltará a BUCLE (JR NZ,BUCLE). (Ver Rutina 2.)

El cargador BASIC del c.m. es:

```
10 CLEAR 49999: LET N=50000:
   FOR H=0 TO 39: READ A:
   POKE N+H,A: NEXT H
20 DATA 33,141,92,54,79,62,0
   ,205,155,34,62,2,205,1,22
   ,6,24,205,68,14,6,0,62,22
   ,215,62,10,215,120,215,62
   ,63,215,4,62,32,184,32,
   239,201
```

Para ejecutar la rutina RANDOMIZE USR 50000

RUTINA NUMERO 3

Hasta ahora hemos utilizado RST #10 para imprimir caracteres en pantalla, pero cuando tenemos una cadena larga de símbolos que representar la rutina se vuelve «pesada» y muy «repetitiva». Para evitarlo utilizaremos una rutina de impresión más fácil de usar y más coherente es la #203C.

Modo de empleo:

- Se borra la pantalla y se abre el canal 2

- En el registro BC se carga la longitud del mensaje a imprimir, incluyendo como parte del mensaje a contar los caracteres de control

- En el registro doble DE se carga la dirección de la memoria donde se almacena el mensaje (se suele poner

TABLA 2

```
                        0    Flash 0    (byte 7)
01001111 ------> 1    Brillo 1   (byte 6)
^        ^              001  Papel 1    (bytes 3,4,5)
.        .              111  Tinta 7    (bytes 0,1,2)
.        byte 0
.
byte 7
```

RUTINA 2

```
10         ORG  50000     ;Colocacion del C/M
20         ENT  50000     ;Direccion de ejecucion (GENS3)
30         LD   HL,23693  ;Variable del sistema ATTR-P
40         LD   (HL),79   ;Mete en ATTR-P el numero 79
50         LD   A,0       ;Pone el borde de color negro
60         CALL 8859      ;
70         LD   A,2       ;Abre el canal de pantalla
80         CALL #1601     ;
90         LD   B,24      ;Borra 24 lineas hacia arriba (toda la
100        CALL #0E44     ;pantalla)
110        LD   B,0       ;Pone el contador a 0
120 BUCLE  LD   A,#16     ;Equivalente a AT
130        RST  #10       ;
140        LD   A,10      ; AT 10,
150        RST  #10       ;
160        LD   A,B       ; AT 10,B  .B varia de 0 a 32
170        RST  #10       ;
180        LD   A,"?"     ;Es lo mismo que poner el valor  ASCII
190        RST  #10       ;del caracter a imprimir
200        INC  B         ;INcrementa b en una unidad
210        LD   A,32      ;Carga A con 32 y lo compara con B
220        CP   B         ;Para comparar hace A-B . Si la  resta
230        JR   NZ,BUCLE  ;da distinto de 0 va a BUCLE. Si A-B=0
240        RET            ;se activa Z y va a RET (va al Basic)
```

RUTINA 3

```
10         ORG  40000     ;Origen
20         ENT  40000     ;Lugar de ejecuccion (solo con el GENS3)
30         CALL #0D6B     ;Borrado
40         LD   A,2       ;Abre el canal de pantalla
50         CALL #1601     ;
60         LD   BC,65     ;Longitud del texto (lineas 100 a 150)
70         LD   DE,MENS   ;Direccion del texto
80         CALL #203C     ;Ejecuccion de la rutina
90         RET            ;Retorno al Basic
100 MENS   DEFB #10,6,#11,1,#12,1,#13,1,#16,10,8 ; tinta 6, papel
                                    azul, flash 1, brillo 1, AT 10,8
110        DEFM "                   " ;
120        DEFB #16,11,8              ;AT 11,8
130        DEFM " INPUT SINCLAIR " ;
140        DEFB #16,12,8              ;AT 12,8
150        DEFM "                   " ;
```

una etiqueta). Los caracteres de control se almacenarán en DEFB y el texto en DEFM (Define *message*). Son, así decirlo las DATAS del BASIC. (Ver Rutina 3.)

El cargador BASIC es:

```
10 CLEAR 39999: LET N=40000:
   FOR H=0 TO 82: READ A:
   POKE N+H ,A: NEXT H
20 DATA 205,107,13,62,2,205,
   1,22,1,65,0,17,82,156,205
   ,60,32,201,16,6,17,1,18,1
   ,19,1,22,10,8,32,32,32,32
   ,32,32,32,32,32,32,32,32,
   32,32,32,32,22,11,8,32,73
   ,78,80,85,84,32,83,73,78,
   67,76,65,73,82,32,22,12,8
   ,32,32,32,32,32,32,32,32,
   32,32,32,32,32,32,32
   ,32
```

Para ejecutar la rutina RANDOMIZE USR 40000.

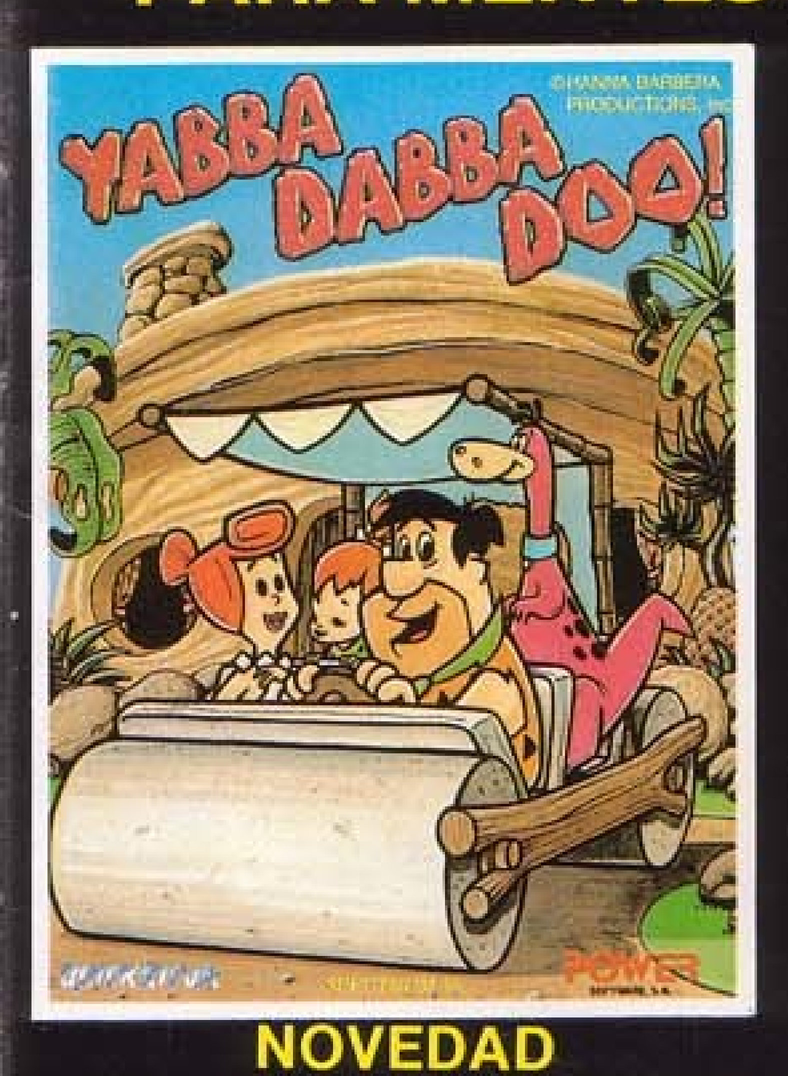
POWER

SOFTWARE, S.A.

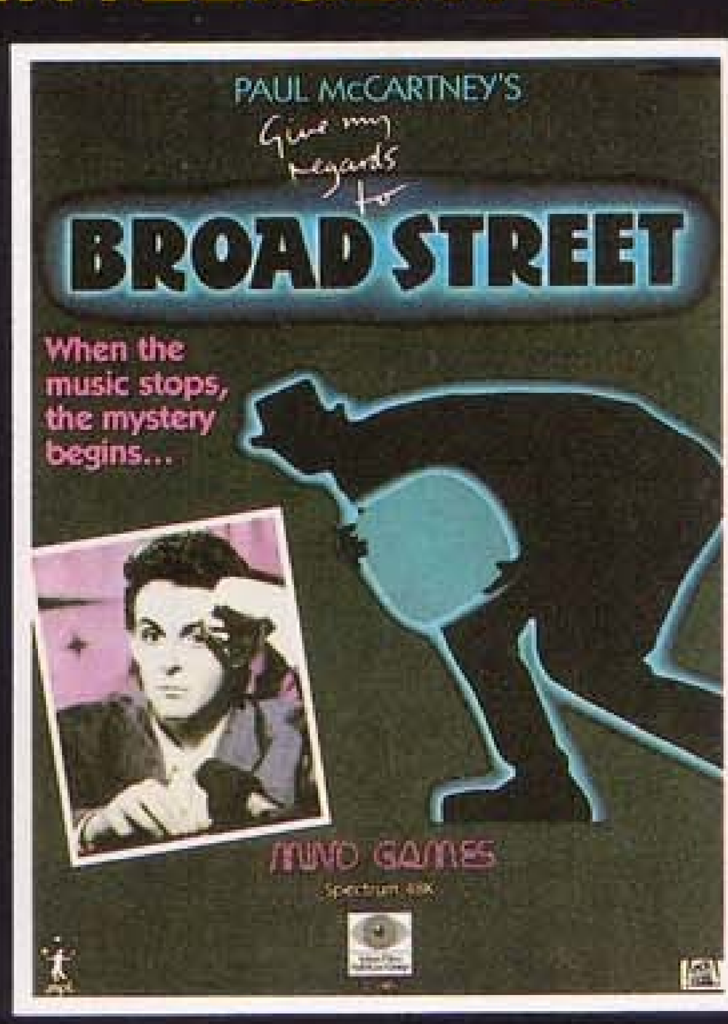
NAPOLES, 98, 1.° 3.ª - Tels. 232 24 61 - 232 25 52

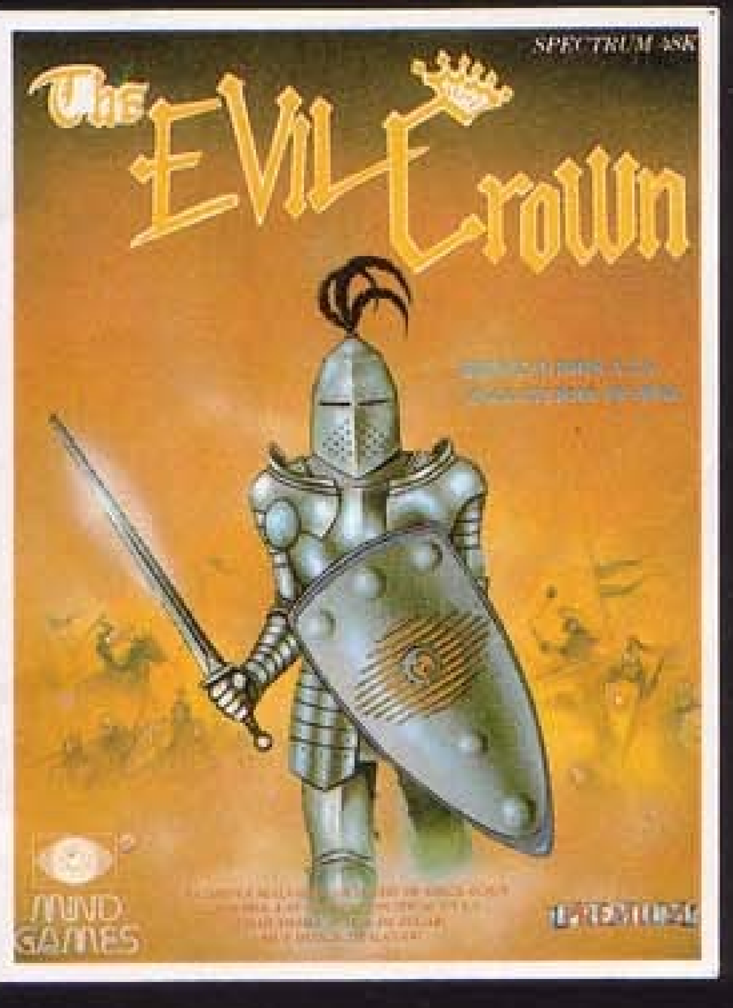
08013 BARCELONA (SPAIN)

JUEGOS INTELIGENTES PARA MENTES INTELIGENTES



NOVEDAD



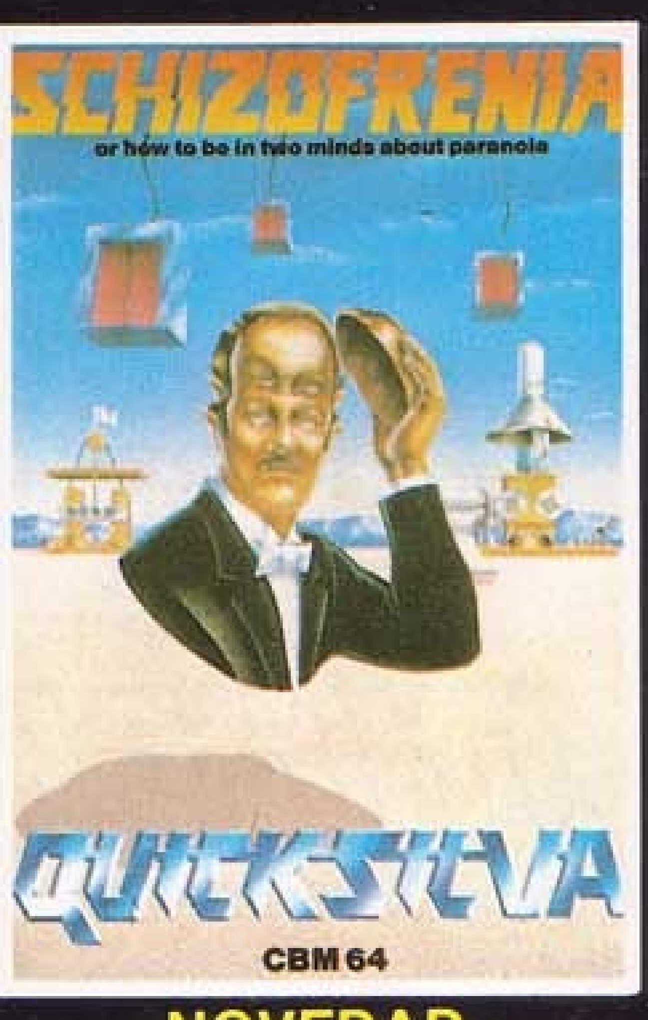




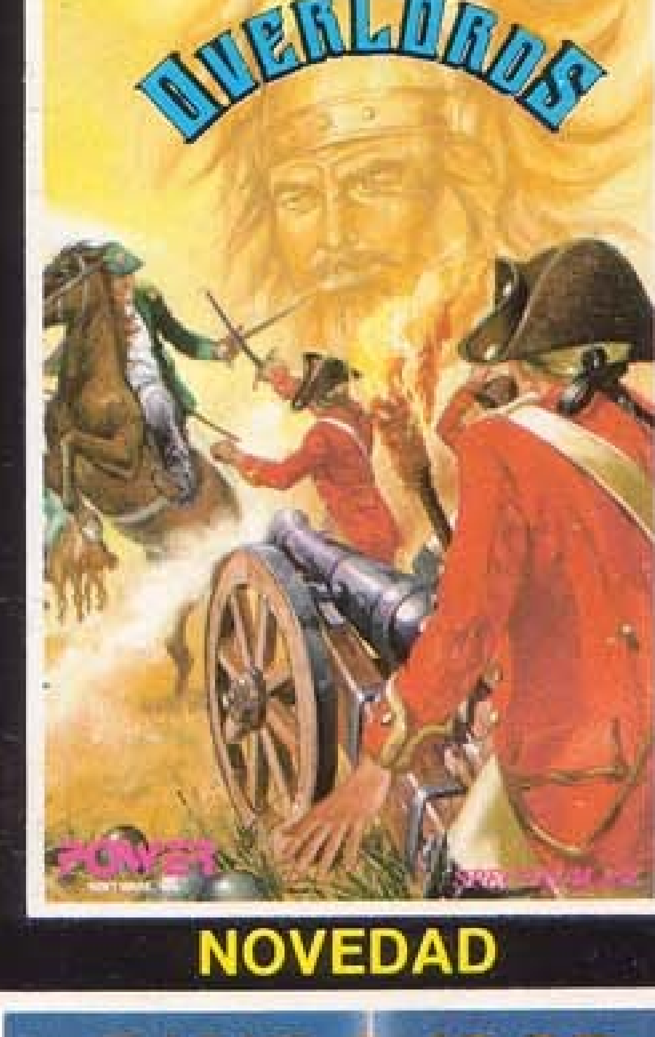


SPECTRUM

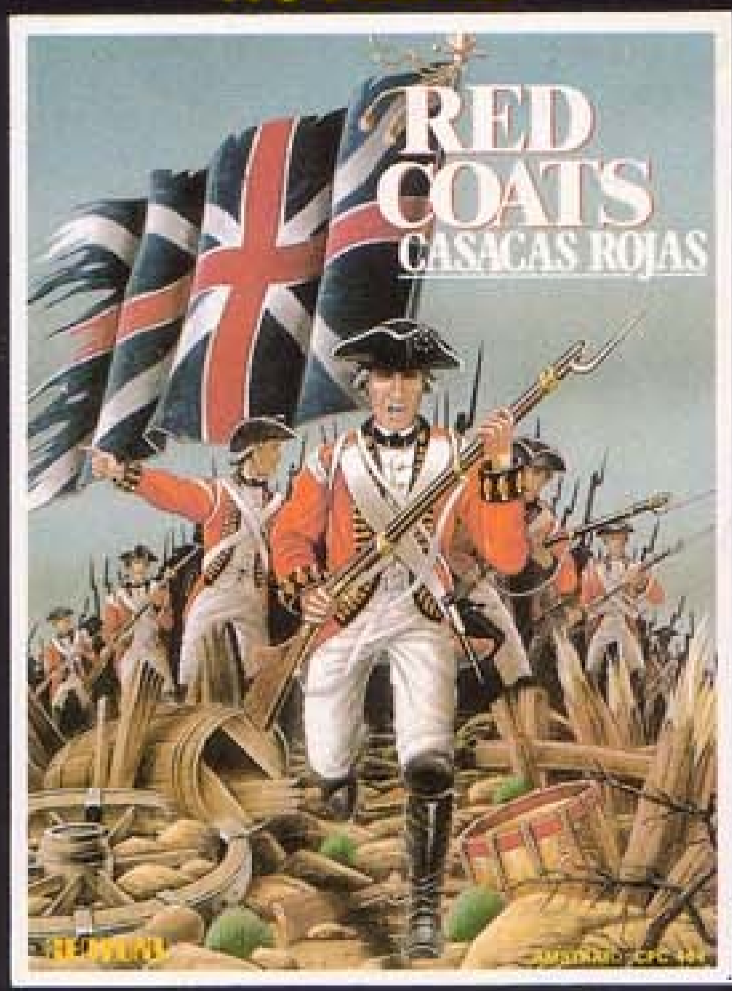
BROAD STREET	2.400 P.V.P.
EVIL CROWN	2.400 P.V.P.
TREASURE ISLAND	2.400 P.V.P.
YABBA DABBA DOO	1.800 P.V.P.
RED COATS	1.800 P.V.P.
SCHIZOFRENIA	2.400 P.V.P.
OVERLORD	1.800 P.V.P.
ANT-ATTACK	1.700 P.V.P.
FALL OF ROME	1.800 P.V.P.
GATECRASHER	1.400 P.V.P.



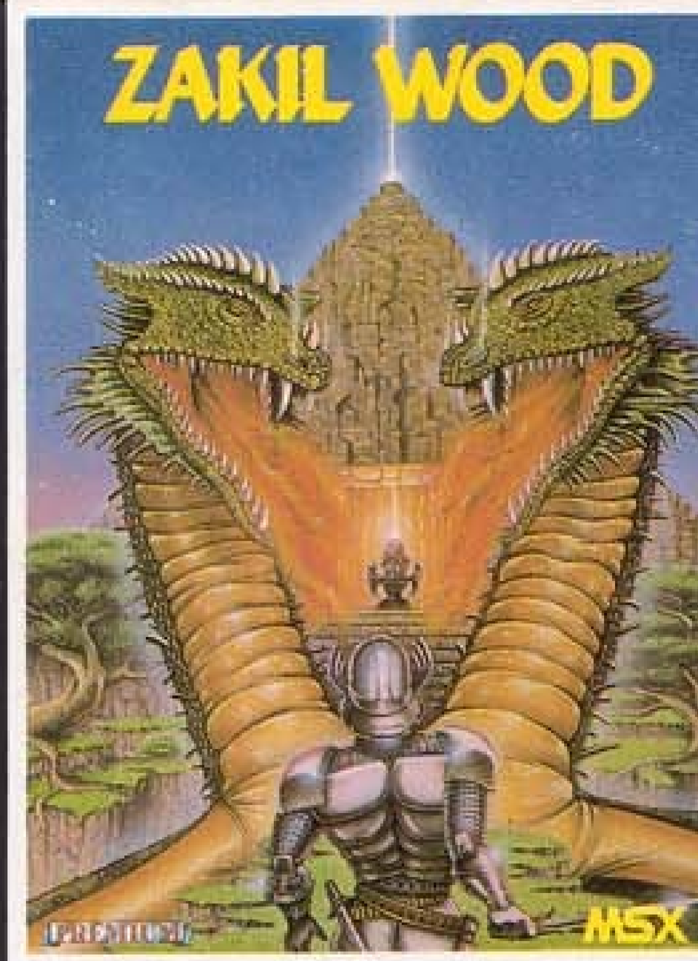
NOVEDAD



NOVEDAD



NOVEDAD



NOVEDAD

COMMODORE

BROAD STREET	Cas. 2.600 – Disc: 3.600 P.V.P.
EVIL CROWN	Cas. 2.600 – Disc: 3.600 P.V.P.
TREASURE ISLAND	Cas. 2.600 – Disc: 3.600 P.V.P.
YABBA BADDA DOO	Cas. 1.800 – Disc: 3.100 P.V.P.
SCHIZOFRENIA	Cas. 2.600 – Disc: 3.600 P.V.P.
ANT ATTACK	Cas. 1.700 – Disc: 2.800 P.V.P.
FALL OF ROME	Cas. 1.800 – Disc: 2.800 P.V.P.
SEE SAW	Cas. 1.600 P.V.P.
TRASHMAN	Cas. 1.600 P.V.P.

AMSTRAD

YABBA DABBA DOO	Cas. 1.800 – Disc. 3.100 P.V.P.
RED COATS	Cas. 2.100 – Disc. 3.100 P.V.P.

M.S.X.

ZAKIL WOOD	Cas. 2.800 P.V.P.
GAMES DESIGNER	Cas. 2.400 P.V.P.

Deseo recibir los juegos que a continuación especifico, comprometiéndome al pago del importe de los mismos

Nombre ______

Dirección ______

Teléfono ______ Ciudad ______

Firma:

Título ______ Sistema ______ Precio ______

Título ______ Sistema ______ Precio ______

Título ______ Sistema ______ Precio ______

Título ______ Sistema ______ Precio ______

☐ Contrarrembolso. ☐ Adjunto Talón. ☐ Giro Postal.

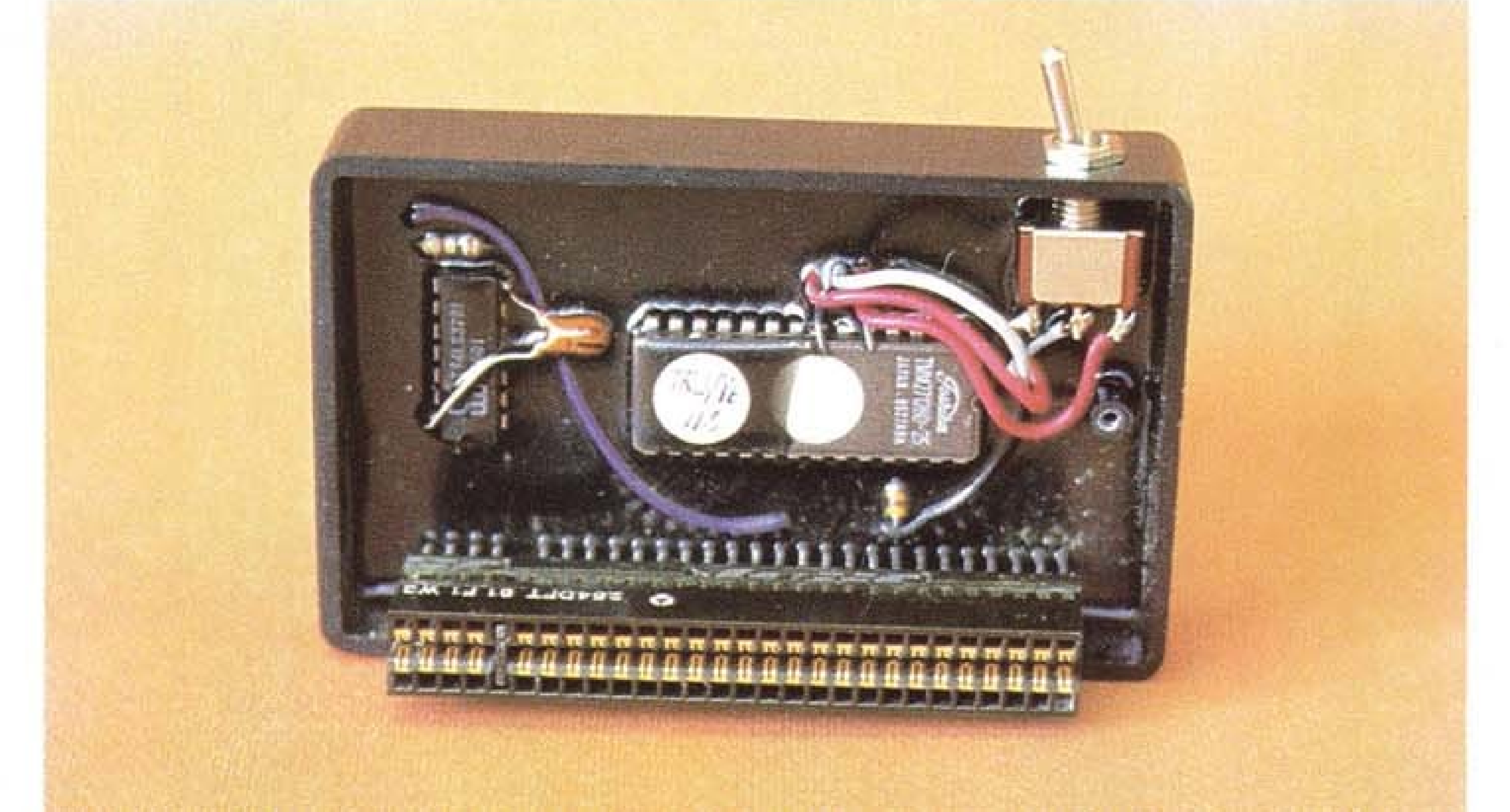
Deseo recibir información de sus programas en: MSX ☐ MASTRAD ☐

INTERFACE TRON

TRON es un *interface* que sirve para *destokenizar* el Spectrum. Con él podemos teclear letra a letra cada palabra clave, como ocurre con «otros ordenadores». Añade además la posibilidad de renumeración automática de las sentencias, caracteres españoles, varios tipos de *hardcopy* (copias por impresora) con protocolo Epson vía RS-232, y otras facilidades.

Es un artilugio muy curioso en lo que a *hardware* se refiere. Cuando está activado «sustituye» la ROM del equipo por su propio sistema operativo (leyendo la ROM del *interface*), siendo éste totalmente compatible con el anterior, con la particularidad de poder pasar de uno a otro sin más que mover el commutador, no deteriorando la tarea que esté efectuando el **Spectrum** en ese momento.

El aspecto físico es muy similar al del *interface* **PHOENIX II**. Consiste en una caja negra de material plástico de tamaño reducido, que en su cara anterior tiene un conector que se adapta al bus de expansión del **Spectrum**. En la posterior tiene el conector de salida del bus, para conexión con otros periféricos. En la cara superior posee un diodo LED para indicar si está en uso el *interface*, y además el commutador que permite activarlo o desactivarlo a voluntad.



La forma de trabajar del **TRON** no es difícil de entender. Cuando está activado «sustituye» parte de las rutinas de la ROM del **Spectrum**, (en especial la de edición de líneas), cambia el juego de caracteres redefiniendo la mayor parte de ellos (parece que tenemos las letras en «negrita») y sustituye la tabla de mensajes de error (aparecen traducidos al castellano).

A modo de ejemplo veamos parte de su funcionamiento. Proponemos el caso de escribir PRINT. Normalmente con pulsar la tecla P nos basta y sobra. Con el *interface* **TRON** conectado, para obtener el mismo resultado necesitamos teclear P+R+I+N+T.

La originalidad del *interface* podemos afirmar que radica en su rutina de edición en código máquina, la cual funciona por medio de interrupciones, detectando y reflejando en las líneas de edición las teclas que estamos pulsando («paralizando» el modo que tiene el **Spectrum** de *tokenizar* las teclas cuando cree que es preciso) hasta que se encuentra con el CHR$(13) (es la tecla ENTER) en cuyo caso *tokeniza* todo lo que se encuentra en la memoria de edición, detecta los posibles errores de sintaxis y, en caso de no existir, envía dicha información al programa o ejecuta la secuencia de comandos, dependiendo en cada caso de la existencia o no del número de sentencia.

Aunque las principales características vienen en el manual de instrucciones, las indicamos a continuación, pues al ser un *interface* de reciente aparición en el mercado es posible que no lo conozcan en los comercios que frecuentéis.

Para obtener los caracteres castellanos hay que seguir una serie de pasos de forma similar a lo que venimos haciendo para obtener los GDU (Gráficos Definidos por el Usuario).

á = E.M. + A
é = E.M. + E
í = E.M. + I
ó = E.M. + O
ú = E.M. + U
ü = E.M. + S.S. + U
ñ = S.S. + U
Ñ = S.S. + Y
¿ = S.S. + X
¡ = S.S. + 2

E.M. = *extend mode*
S.S. = *symbol shift*

Además de estos caracteres castellanos, el **TRON** añade al BASIC cuatro comandos nuevos, tres de ellos dedicados a la manipulación de programas y el restante para facilitarnos las tareas de *hardcopy* (copia de la pantalla en alta resolución en papel) mediante el *interface* serie RS-232 con protocolo **Epson**, muy conocida gracias a su difusión con los **PC**'s de **IBM**). Estos comandos son:

FORMAT n1,n2 renumera los números de sentencia comenzando por el n1 y contando de n2 en n2.

ERASE n1,n2 borra las sentencias comprendidas entre n1 y n2, ambas inclusive.

CAT ı, similar al comando **FREE()** del **BASIC MICROSOFT**, indica el número de bytes libres de memoria RAM disponibles para el usuario.

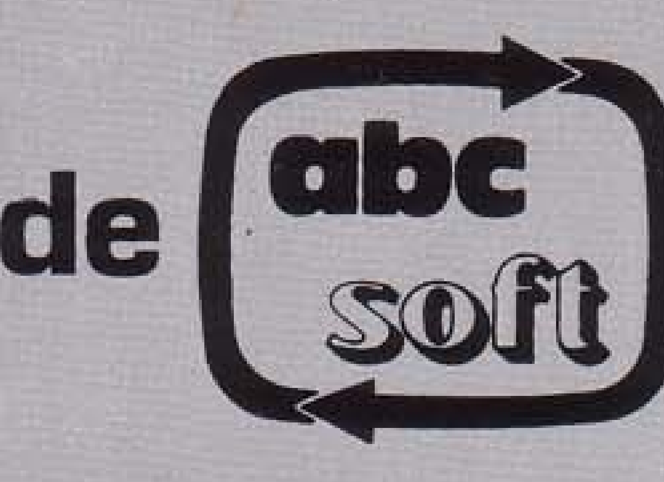
MOVE COPY n, donde n puede valer 0,1,2 ó 3 según queramos obtener un *hardcopy* normal, ampliado, en inversa o en inversa ampliado.

Tanto estos nuevos comandos como los anteriores (los propios del **Spectrum**) deben escribirse en mayúsculas.

Como nota importante hay que indicar que el **TRON** tiene que ser el primero de todos los *interfaces* que conectemos al **Spectrum**, cosa que resulta bastante «engorrosa» en el caso de utilización del **Discovery**.

El **TRON**, al igual que el **PHOENIX II-E**, ha sido diseñado por la empresa madrileña **ABACO**. Su P.V.P. viene a ser de unas 7.000 ptas. (sin IVA).



de




Estos programas son los más vendidos en el Reino Unido, según las encuestas de ventas realizadas por GALLUP/MICROSCOPE, en las semanas del 10 de Marzo al 12 de Abril de 1986.

SPECTRUM

TITULO	EDITOR	P.V.P.R.
1 ( ) [3] GREEN BERET	IMAGINE	2.100
2 ( ) [4] WAY OF THE TIGER	GREMLIN	2.100
3 ( ) [3] BOMB JACK	ELITE	2.575
4 ( ) [5] TURBO SPRIT	DURELL	2.100
5 ( ) [6] INCREDIBLE SHRINK	MASTERTRONIC	750
6 ( ) [3] SUPERBOWL	OCEAN	2.800
7 ( ) [7] F A CUP FOOTBALL	VIRGIN	2.100
8 ( ) [2] V	OCEAN	2.100
9 ( ) [10] MOVIE	IMAGINE	2.100
10 ( ) [8] DEVILS CROWN	MASTERTRONIC	750

COMMODORE 64

TITULO	EDITOR	P.V.P.R.
1 ( ) [6] URIDIUM	HEWSON CONS	2.500
2 ( ) [3] SUPERBOWL	OCEAN	2.800
3 ( ) [7] HARDBALL	ACCOLADE	2.300
4 ( ) [2] V	OCEAN	2.300
5 ( ) [11] KANE	MASTERTRONIC	750
6 ( ) [9] YIE AR KUNG FU	IMAGINE	2.300
7 ( ) [4] PING PONG	IMAGINE	2.500
8 ( ) [4] ZAPP SIZZLERS	GREMLIN	2.800
9 ( ) [11] KUNG FU MASTER	US GOLD	2.600
10 ( ) [7] ZOIDS	MARTECH	2.575

AMSTRAD

TITULO	EDITOR	P.V.P.R.
1 ( ) [4] WAY OF THE TIGER	GREMLIMS	2.300
2 ( ) [6] RAMBO	OCEAN	2.300
3 ( ) [3] SPINDIZZY	ELECTRIC DREAM	2.450
4 ( ) [3] LAST V 8	MASTERTRONIC	1.100
5 ( ) [5] COMPUTER HITS V2	BEAU JOLLY	3.400
6 ( ) [30] FINDERS KEEPERS	MASTERTRONIC	750
7 ( ) [2] INTO OBLIVION	MASTERTRONIC	750
8 ( ) [20] FORMULA ONE SIM.	MASTERTRONICS	750
9 ( ) [18] THEY SOLD A MILL.	HIT SQUAD	2.500
10 ( ) [6] ONE MAN AND HITS.	MASTERTRONIC	750

COMMODORE 16

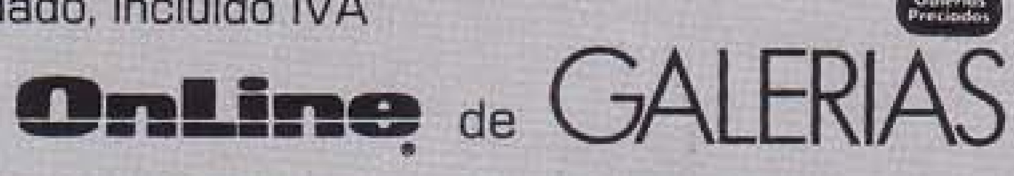
TITULO	EDITOR	P.V.P.R.
1 ( ) [8] MR. PUNIVERSE	MASTERTRONIC	750
2 ( ) [6] COMMANDO	ELITE	2.575
3 ( ) [10] BIG MAC	MASTERTRONIC	750
4 ( ) [14] FORMULA ONE SIM.	MASTERTRONIC	750
5 ( ) [6] BMX RACER	MASTERTRONIC	750

MSX

TITULO	EDITOR	P.V.P.R.
1 ( ) [6] FORMULA ONE SIM.	MASTERTRONIC	750
2 ( ) [5] FINDERS KEEPERS	MASTERTRONIC	750
3 ( ) [3] WAY OF THE TIGER	GREMLIMS	2.300
4 ( ) [4] CHILLER	MASTERTRONIC	750
5 ( ) [1] TURMOIL	BUG BYTE	1.200

( ) Posición en mes anterior, [ ] n.º de semanas entre los más vendidos.

P.V.P.R.: Precio de venta recomendado, incluido IVA

* Encuéntralos en la división


* Y en comercios de microinformática

* Es una promoción de:

abc analog, s. a.


Santa Cruz de Marcenado, 31 (3.º 14) 28015 MADRID

Tfnos: (91) 248 82 13 - 242 50 59
Télex. 44561 BABC E

INTERFACE PHOENIX II-E

El **Phoenix II-E** es el sucesor de la línea iniciada con el **Phoenix** al que se le han efectuado algunas modificaciones. Se le ha reducido el tamaño (resultaba muy aparatoso) y se le ha eliminado el botón de *reset* (según la casa diseñadora es una duplicación innecesaria porque los últimos modelos presentados por el **Sr. Sinclair** incluyen el interruptor de *reset*). Funciona sin problemas en el **Spectrum** de 48K, **Plus** y **128** en modo 64K, pues, aunque muchos no lo crean, son el mismo microordenador.

El manejo es muy sencillo, y no requiere conocimientos de código máquina ni la utilización de programas copiadores. Con el **Spectrum** desenchufado, se inserta el *interface* en la ranura de expansión y se procede a la carga del programa a duplicar con total normalidad. Una vez hecho esto apretamos el pulsador situado en la parte superior, empezando a funcionar su sistema operativo, con lo que veremos que el color del borde se ve sustituido por unas franjas negras y verdes en movimiento descendente. Ya estamos en condiciones de utilizar el *interface*.

Para copiar un programa debemos pulsar la tecla 1, poniéndose el borde de color azul, y una vez desconectada la clavija EAR y puesto el *cassette* en modo de grabación, podemos elegir una de las siguientes opciones:

1 - Para poderlo pasar después a **Microdrive** o **Discovery**
2 - Copia a velocidad normal
3 - Copia turbo a 2000 baudios
4 - Copia turbo a 2500 baudios
5 - Copia turbo a 3000 baudios

La forma de trabajar del **Phoenix** es muy sencilla. Una vez pulsado el botón del *interface* se vuelca una parte de *software* en el primer tercio de la memoria de pantalla, «congelando» el contenido de la memoria y fotografiando por así decirlo, las variables del sistema, para poder recomenzar el programa copiando en la situación en que se dejó. Una vez elegida la opción deseada, se va volcando la memoria RAM en el *cassette* de una forma troceada, generándose a su vez el programa cargador adecuado para cada opción.



Hay tres detalles que podemos destacar como claras ventajas:

- Para utilizar las copias obtenidas con el *interface* no es necesario tenerlo conectado.

- Nos ofrece la posibilidad de pasar nuestros programas favoritos a **Microdrive** o **Discovery** (los usuarios del *interface* **Betadisk** tienen que hacer una pequeña modificación del programa cargador).

- Podemos pasar a velocidad normal los originales comprados con protección turbo y que tanto nos cuesta cargar.

Para todas las personas que se aprovechan del trabajo de otros, y compiten deslealmente con los vendedores de *software*, los vulgarmente llamados «piratas», no creemos que les pueda ser de mucha utilidad, ya que el tiempo de carga es elevado al tener que grabar toda la memoria RAM. Además la pantalla queda ligeramente alterada dando muestras de no ser un original sino una simple copia de trabajo. No te dejes engañar si te venden un programa con la pantalla manipulada.

Como notas reseñables hay que destacar la posible dificultad de lectura de los programas grabados con las opciones 4 y 5, requiriéndose quizá la utilización de *cassettes* de mayor calidad. También hay que señalar lo engorroso que resulta en la mayoría de los casos pasar programas al **Discovery**, teniendo que desenchufar el **Spectrum** de la unidad de disco, proceder según las instrucciones y una vez obtenida la copia en cinta, volvemos a conectarlo al **Discovery** cargando ya el programa sin problemas.

La empresa diseñadora y distribuidora del **Phoenix II-E** es el **Grupo Abaco Informática**. El precio de este *interface* en la calle es de 8900 ptas.+12% IVA. Como valoración final podemos decir que es un *interface* interesante y que puede potenciar el uso de nuestra unidad de discos.

La rutina gráfica que te permitirá presentar las cartas en la pantalla, sería algo así:

```
10 BORDER 4: PAPER 4: INK 9:
   CLS: POKE 23658,8: LET B
   =0: LET C=1
20 FOR N=USR "A" TO USR "R"+
   7: READ A: POKE N,A:
   NEXT N
30 DIM C(52): FOR N=C TO 52:
   LET C(N)=N: NEXT N
40 DIM A(13,13,2)
50 FOR N=C TO 10: FOR M=C TO
   N: READ A(N,M,C),A(N,M,2)
   : NEXT M: NEXT N
60 FOR N=11 TO 13: LET A(N,C
   ,C)=4: LET A(N,C,2)=2:
   NEXT N
70 LET CC=C: LET CP=100
80 GO SUB 5000
500 LET Y=0: LET X=1
525 LET Z=C(CC)
530 GO SUB 5500
540 STOP
5000 CLS: PRINT AT 10,10;
     "BARAJANDO"
5010 FOR N=C TO 100
5020 LET X=INT (RND*52)+C
5030 LET Y=INT (RND*52)+C
5040 LET Z=C(X): LET C(X)=C
     (Y): LET C(Y)=Z
5050 NEXT N
5060 CLS: RETURN
5500 FOR N=Y TO Y+8: PRINT
     PAPER 7; AT N,X; "[5*
     ESPACIO]": NEXT N
5510 LET ST=INT ((Z-C)/13)
5520 LET CH=144+ST
5530 LET VA=Z-(13*ST)
5540 IF ST<2 THEN INK 2
5560 LET AC=147+VA
5600 PRINT PAPER 7; AT Y,X;
     CHR$ AC; AT Y,X+4; CHR$
     AC; AT Y+8,X; CHR$ AC;
     AT Y+8,X+4; CHR$ AC
5610 FOR N=C TO VA: IF A(VA,
     N,C)<>B THEN PRINT
     PAPER 7; AT Y+A(VA,N,C)
     ,X+A(VA,N,2); CHR$ CH
5620 NEXT N
5890 INK 9
5900 RETURN
9000 DATA 0,54,127,127,127,
     62,28,8,0,8,28,62,127,
     62,28,8,8,28,62,127,127
     ,62,8,28,8,28,28,107,
```



```
        127,107,8,28
9010 DATA 0,8,20,34,34,62,34
        ,34,0,28,34,2,4,24,32,62
9020 DATA 0,28,34,2,12,2,34,
        28,0,4,12,20,36,62,4,14
9030 DATA 0,62,32,32,60,2,34
        ,28,0,28,34,32,60,34,34
        ,28
9040 DATA 0,62,34,2,4,8,16,
        16,0,28,34,34,28,34,34
        ,28
9050 DATA 0,28,34,34,30,2,34
        ,28,0,76,82,82,82,82,82
        ,76
9060 DATA 0,14,4,4,4,4,36,24
        ,0,28,34,34,34,58,102,
        30,0,118,36,40,48,40,36
        ,118
9070 DATA 85,85,85,85,85,85,
        85,85
9100 DATA 4,2
9110 DATA 2,2,6,2
9120 DATA 2,2,4,2,6,2
9130 DATA 1,1,1,3,7,1,7,3
9140 DATA 1,1,1,3,4,2,7,1,7
        ,3
9150 DATA 1,1,1,3,4,1,4,3,7,
        1,7,3
9160 DATA 1,1,1,3,2,2,4,1,4,3
        ,7,1,7,3
9170 DATA 1,1,1,3,2,2,4,1,4,3
        ,6,2,7,1,7,3
9180 DATA 1,1,1,3,3,1,3,3,4,2
        ,5,1,5,3,7,1,7,3
9190 DATA 1,1,1,3,2,2,3,1,3,3
        ,5,1,5,3,6,2,7,1,7,3
```

La forma de trabajo del programa es la siguiente:

Los colores del borde de pantalla, el papel y la tinta se establecen en la línea 10, mientras que el POKE pone la máquina en modo de mayúsculas. En todo lo que sigue se utilizan las dos variables B y C en lugar de los números 0 y 1. Debido a la forma en que el **Spectrum** almacena los números y las variables puedes ahorrar seis bytes de memoria cada vez, lo que hace posible que el programa corra en el **Spectrum** de 16K.



En la línea 20 se definen los GDUs de los símbolos utilizados para los cuatro palos, así como los GDUs de los números y letras que se utilizan en las esquinas de las cartas. Los correspondientes valores de los DATA se almacenan en las líneas 9000 a 9070. A continuación la línea 30 define un mazo de cartas sin barajar. La matriz A, dimensionada en la línea 40, se llena con los datos de las líneas 9100 a 9190, que da las cordenadas de las figuras de cada carta. La línea 50 llena una parte de la matriz con las posiciones de los símbolos de los palos en las cartas. La línea 60 llena el resto de la matriz con la posición de los símbolos de las figuras. Se podrían dibujar unas figuras más adecuadas para representar las cartas, pero resultaría muy tediosa la introducción de todos los datos, aparte de que difícilmente entrarían en una máquina de 16K.

En la línea 70 se pone CC a 1 y CP a 100. CC es la carta activa en un momento dado, es decir, el elemento de la matriz de naipes que está siendo manejado por el programa. CP es el número de «piedras» que hay en posesión del jugador.

En la línea 80 se llama a la subrutina que baraja las cartas y empieza en la línea 5000. La pantalla se borra y en la misma línea 5000 se presenta mensaje correspondiente. Para barajar se eligen dos cartas al azar y se permutan sus posiciones. El bucle FOR ... NEXT, que empieza en la línea 5010 sirve para comprobar que se han hecho 100 permutaciones. Según el cálculo de probabilidades, puede ocurrir que algunas veces se elija la misma carta en las líneas 5020 y 5030, con lo que dicha carta resultará permutada consigo misma. Realmente esto no importa mucho, ya que 100 permutaciones suponen un barajado bastante completo. Si no te parece suficiente la cifra de 100 permutaciones, puedes modificar el valor que aparece en la línea 5010, pero muy pronto te irás a tiempos absurdamente largos.

La subrutina termina en la línea 5060 que borra la pantalla y hace un RETURN.



La línea 525 asigna a la variable Z el valor de la carta activa de la matriz C, antes de que desde la línea 530 se llame a la subrutina que empieza en la línea 5500. Esta subrutina está relacionada con la representación de las cartas en la pantalla. La línea 5500 visualiza las cartas; de momento sólo presenta el fondo blanco, los símbolos y los números se encargará de visualizarlos el resto de la subrutina. La línea 5510 establece de qué palo será la carta elegida; cada palo tiene asignado un número del cero al tres. La línea 5520 decide la cadena de caracteres que corresponde al GDU de ese palo en particular.

Finalmente, querrás saber cuál es el número de la carta dentro de su palo: ¿será un as, un dos, o tal vez una dama?

Antes de representar en el naipe los símbolos de los palos, hay que definir el color de la tinta. La línea 5540 asigna el color rojo cuando el número del palo es cero o uno, y el color negro en los otros dos casos.

La línea 5600 dibuja el número de la carta, haciendo uso del GDU adecuado, que toma de los definidos anteriormente. Los símbolos se presentan por medio de la línea 5610. Las coordenadas de cada símbolo se sacan de la matriz A, presentándose los GDUs en esa posición.

Por último se define INK9, la tinta de contraste y se produce el RETURN.

EMPIEZA EL JUEGO

El banquero fija en tí su mirada de hielo. Haces tu apuesta y recibes otra carta, pero ¿doblas, pides carta o te plantas? En esta ocasión nos ocuparemos de las líneas de programa correspondientes a la parte del jugador.

Continuando a partir de la rutina de gráficos que tecleaste en el capítulo anterior, ahora te hacen falta dos secciones más de programa: una para manejar las respuestas del jugador y la otra para hacer posible que pueda jugar el ordenador. En este capítulo veremos las líneas que hacen falta para el jugador, y además a las «piedras» les llamaremos fichas. Pero no llegarás muy lejos jugando ahora, ya que todavía no has enseñado a jugar a tu ordenador.





- REPARTIENDO LAS CARTAS
- LAS APUESTAS
- DOBLAR, CAMBIAR Y PLANTARSE
- HACER SALTAR A LA BANCA
- EJECUCION DE LOS TOTALES

Esta sección del programa está relacionada con unas cuantas tareas diferentes. (No te preocupes si no estás muy seguro de las reglas exactas del juego de las veintiuna, ya nos ocuparemos de ellas junto con la última parte del programa). Básicamente el programa tiene que hacer tres cosas. Tiene que ocuparse del reparto de las cartas, permitir al jugador hacer apuestas y pedir cartas adicionales, y finalmente tiene que calcular la puntuación del jugador.

En la siguiente sección de programa, se le pide al jugador que haga una apuesta sobre la primera carta que da el ordenador (la banca), una para el jugador y otra para el propio ordenador. Después de recibir la segunda carta, al jugador se le presentan tres opciones: plantarse, pedir nueva carta o doblar, según el estado del juego.

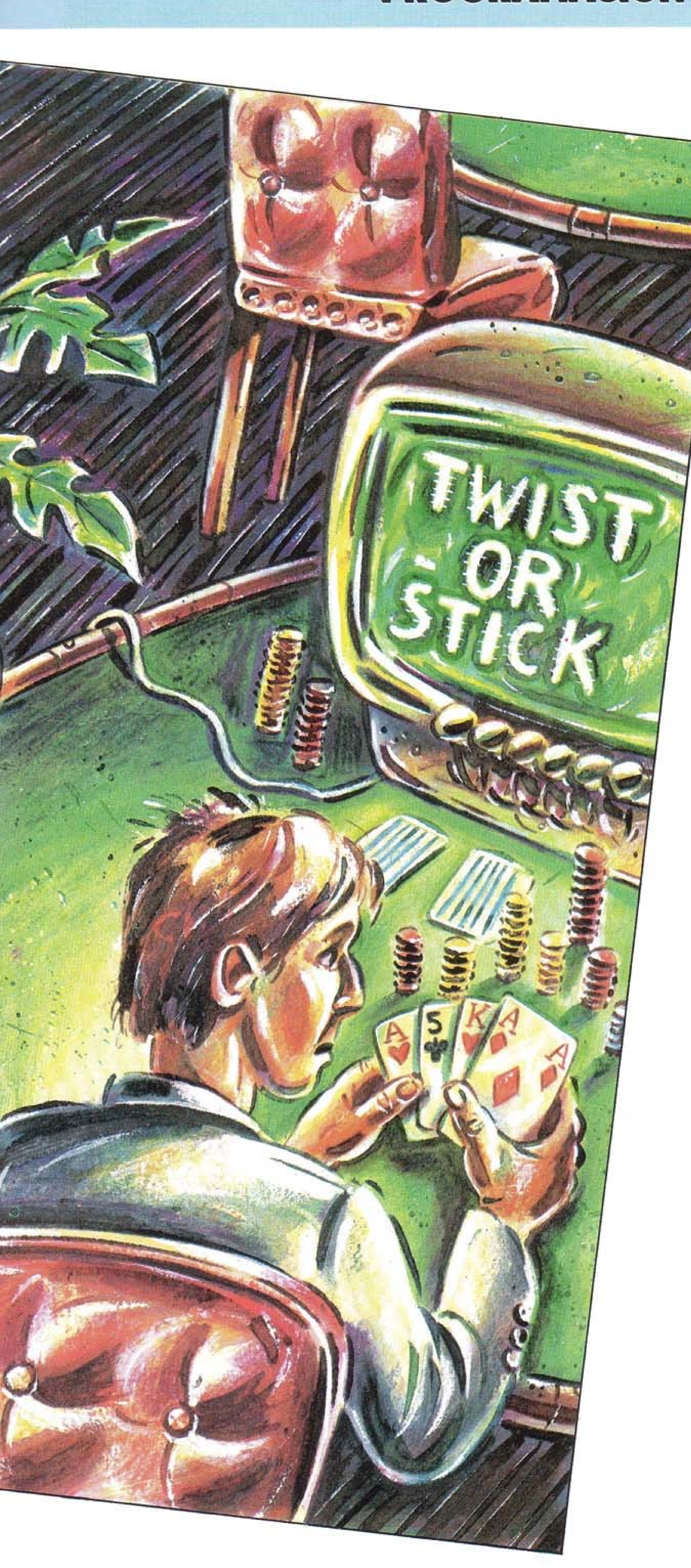
El programa tiene rutinas para manejar las tres opciones, repartiendo cartas extras y/o añadiéndolas en la puesta, o pasar la vez al banquero. Después de cada nueva carta el programa comprueba si la puntuación ha pasado de 21 en cuyo caso el jugador pierde.

El programa comprueba además las condiciones especiales: las veintiuna y las cinco cartas. Si se produce alguna de ellas, aparece un mensaje que dice al jugador la puntuación obtenida y pasa el turno a la banca.

Agrega ya las siguientes líneas al programa. La línea 530 se ha repetido, tomándola de la vez anterior:

```
90 DIM S(2): LET BET=B
100 DIM O(2): DIM W(2)
500 LET Y=B: LET X=C: LET TF
    =B: LET AF=B: LET PF=B:
    LET FF=B
520 FOR U=C TO 2
530 GO SUB 5500: GO SUB 6000
    : GO SUB 6500
535 IF CC=52 THEN LET CC=B
540 LET O(U)=C(CC+C)
550 LET X=X+6
560 PRINT PAPER 7; AT 21,B;
    "TIENES ";CP;" FICHAS "
570 IF U=C THEN INPUT "CUAL
    ES TU PUNTUACION? ";AM:
    IF AM>CP OR AM<C THEN GO
    TO 570
590 IF U=C THEN LET CP=CP-AM
    : LET BET=AM
595 GO SUB 7000: GO SUB 7000
    : NEXT U
622 IF S(2)=21 THEN LET PF=C
    : PRINT PAPER 2; AT 4,18
    ;" VENTIUNO ":GO TO 2500
700 IF (S(C)<16 AND (S(2)<16
    OR S(2)>21) OR TF=C OR
    CP<AM) THEN GO TO 705
702 INPUT "(D)OBLAS, (C)ARTA
    O (T)E PLANTAS? "; LINE
    D$: IF D$<>"D" AND D$<>
    "C" AND D$<>"T" THEN GO
    TO 702
703 GO TO 715
705 IF (S(C)>21 OR CP<AM OR
    TF=C) THEN GO TO 710
706 INPUT "(D)OBLAS O
    (C)ARTA? ";LINE D$: IF
    D$<>"D" AND D$<>"C" THEN
    GO TO 706
708 GO TO 720
710 IF (S(C)<16 AND (S(2)<16
    OR S(2)>21)) THEN GO TO
    770
711 INPUT "(C)ARTA O (T)E
    PLANTAS? "; LINE D$: IF
    D$<>"C" AND D$<>"T" THEN
    GO TO 711
715 IF D$="T" THEN GO TO 2500
720 IF LEN D$=B THEN GO TO
    700
725 IF D$(C)<>"D" THEN LET
    TF=C: GO TO 905
760 LET BET=BET+AM: LET CP=
    CP-AM
770 PRINT PAPER 7; AT 21,B;
    "TIENES ";CP; " FICHAS"
905 LET Z=C(CC)
910 GO SUB 5500: GO SUB 6000:
    GO SUB 7000
920 LET X=X+6
921 IF S(C)>21 THEN GO TO
    2000
925 IF X=31 AND S(C)<22 THEN
    PRINT PAPER 2; AT 21,B;
    " HAS HECHO JUEGO DE 5
    CARTAS! ": LET FF=C: GO
    TO 2500
```


BET
NOW




```
930 GO TO 700
2000 IF CP=B THEN PRINT AT
     21,B; "HAS PERDIDO TODO
     !": STOP
2010 IF CP>=1000 THEN PRINT
     AT 21,B; "FELICIDADES!
     HAS DESBANCADO AL
     ORDENADOR!!": STOP
2020 PRINT #C;"PULSA 'S' PARA
     OTRA MANO"
2030 IF INKEY$<>"S" THEN GO
     TO 2030
2037 IF PF=C OR DPF=C THEN
     GO SUB 5000
2040 CLS: GO TO 90
2500 STOP
6000 IF VA>10 THEN LET
     VA=10
6010 LET S(C)=S(C)+VA: LET S
     (2)=S(2)+VA
6020 IF VA=C AND AF=B THEN
     LET S(2)=S(2)+10: LET
     AF=C
6030 IF S(C)>21 THEN PRINT
     FLASH C; PAPER 7; AT
     20,B;" "; TAB 9;"HAS
     FALLADO! ";TAB 31;" ":
     LET DPF=B: RETURN
6040 PRINT AT 20,B;" "; TAB
     31; " "
6050 PRINT PAPER 7; INK B;
     AT 20,B; "TU PUNTUACION
     ES "; S(C);: IF S(2)<>
     S(C) AND S(2)<22 THEN
     PRINT PAPER 7; INK B;
     " O "; S(2)
6060 RETURN
6500 FOR N=10 TO 18
6510 PRINT PAPER 7; INK 1;
     AT N,X; CHR$ 161; CHR$
     161; CHR$ 161: CHR$ 161
     ; CHR$ 161
6520 NEXT N
6530 RETURN
7000 LET CC=CC+C: IF CC=53
     THEN LET CC=C
7010 RETURN
```

Las líneas 90 y 100 definen la variable BET y las tres matrices: S contiene dos totales de puntuación distintos para el jugador (el as puede contar como uno o como once), O contiene las dos cartas iniciales del ordenador y W contiene las puntuaciones de esas

dos cartas (cuando hay un as presente).

MAS CARTAS

La línea 500 define las coordenadas de la esquina superior izquierda de la carta, poniendo a cero cuatro indicadores. TF es el indicador de cambio, que se pone a uno tan pronto como el jugador elige cambiar durante el juego (al jugador se le impedirá entonces comprar cartas), AF es el indicador de as (utilizado para que se calculen dos totales, en los que el as interviene con puntuaciónes 1 y 11), PF es el indicador de las veintiuna, que indica que el jugador ha hecho veintiuna y por último FF es el indicador de las cinco cartas.

Entre las líneas 520 y 595 hay un bucle FOR ... NEXT que se ocupa de repartir las dos primeras cartas al jugador y a la banca. La línea 530 llama entonces a dos subrutinas extras: GOSUB 6000 actualiza la puntuación a medida que se van repartiendo las cartas, mientras que GOSUB 6500 presenta las cartas por el lado opuesto, las cartas de la banca.

Descendiendo más al detalle, la línea 6000 comprueba si la carta corresponde a una figura, en cuyo caso le asigna el valor 10. La línea 6010 suma el valor de los dos elementos de la matriz de puntuación. La comprobación de los ases se hace en la línea 6020. La línea 6030 comprueba si el jugador tiene más de 21 puntos. Si así es, le dice: HAS FALLADO.

Después de volver al cuerpo principal del programa, la línea 535 comprueba si la carta actual es la número 52, cambiándola por la 0 cuando esto ocurre. La línea 540 coloca las cartas del ordenador en la matriz O. La línea 550 desplaza la siguiente carta seis posiciones de pantalla hacia la derecha.

APUESTAS

La línea 560 presenta el número de fichas que el jugador mantiene, mientras que la línea 570 le pide que haga su apuesta, pero sólo después de la primera carta. Dicha línea también se cerciora de que el jugador tiene suficiente número de fichas para la apuesta. La puesta se resta del total de fichas y el valor de la propia variable BET (APUESTA) se asigna en la línea 590. Se necesitan las dos variables, AM y BET para mantener el seguimiento de cómo va comprando cartas el jugador, para el caso de que posteriormente elija esa opción.

Los dos GOSUB 7000 de la línea 595 sirven para ajustar CC, la carta actual, y asegurarse de que CC vuelve al principio del mazo cuando es demasiado grande. Aquí termina el bucle FOR ... NEXT.

La línea 622 comprueba si las dos cartas que se le han repartido al jugador forman las veintiuna. El indicador de las veintiuna, PF, se activa en ese caso y el programa se para. La sección a partir de la línea 2500 se añadirá en el próximo artículo. Las líneas 700, 705 y 710 comprueban diversas combinaciones de puntuación, indicador de cambio y fichas disponibles. Su misión es seleccionar la combinación correcta de opciones que ha de ser presentada al jugador. Por ejemplo, si las puntuaciones están por encima de 16 y por debajo de 21, y además el jugador todavía no ha cambiado, y además tiene suficientes fichas para comprar una carta, entonces la línea 702 le brinda al jugador las opciones de doblar, cambiar o plantarse. Si por otra parte el jugador ya ha cambiado una carta, ya no tiene abierta la opción de doblar, por lo que la línea 711 le presenta las opciones de cambiar o plantarse. La combinación final de opciones es cambiar o doblar si ninguna de las puntuaciones supera los 16 puntos. Estas opciones se presentan en la línea 706. Si el jugador decide plantarse en este punto, termina aquí esta versión corta del programa.

La línea 729 comprueba si el jugador ha pulsado la tecla ENTER en lugar de otra. Caso de ser así el programa vuelve a ofrecer las opciones. La línes 725 pone a uno el indicador de cambio en el caso de que el jugador no haya doblado una carta. El programa salta entonces a la línea 905.

Si el jugador ha decidido doblar una carta, el programa llega a la línea 760, ajusta el valor de BET y el número de piedras CP. La línea 770 presenta el número de fichas que quedan.

Las líneas 905 a 920 presentan en la pantalla las cartas adicionales, las que han sido cambiadas o dobladas. La línea 921 agrega la puntuación. Cuando está por encima de 21, el programa salta a la línea 2000, el lugar donde empiezan las comprobaciones cuando el jugador se pasa.

La línea 925 es la comprobación de las cinco cartas. Esta comprobación se hace examinando la posición de la última carta repartida en la pantalla. Si es la posición de la quinta carta y la puntuación está por debajo de 22, el jugador ha conseguido un triunfo de cinco cartas. El indicador de cinco cartas queda activado y el programa salta a la línea 2500. Si el jugador ha decidido no plantarse y no ha acumulado un triunfo de cinco cartas, la línea 930 vuelve a enviar el programa al la línea 700.

HACIENDO SALTAR LA BANCA

La línea 2000 comprueba si el jugador ha perdido toda su provisión de fichas, informándole cuando así sucede. El juego termina cuando el jugador ya no tiene más. La línea 2010 sirve para identificar un final más feliz: el caso de que el jugador haya acumulado más de 1000 fichas, caso en que el banco quiebra y el juego también termina.

Si el jugador todavía deja algunas fichas, pero no ha logrado acumular más de 1000, la línea 2020 le pregunta si desea jugar otra mano más. PRINT #1 pondrá dicho mensaje en la primera de las dos líneas no utilizables normalmente que hay al fondo de la pantalla.

Cuando el jugador o el que reparte las cartas han conseguido hacer las veintiuna, la línea 2037 se ocupa de barajar las cartas, llamando a la subrutina de barajado. Si el jugador desea jugar otra mano más, la línea 2040 borra la pantalla y empieza de nuevo.

UNIDAD DE DISCO DISCOVERY 1

Siguiendo con nuestra intención de presentar el lado profesional del **ZX-Spectrum** vamos a centrar hoy nuestra atención en la unidad de discos **Discovery 1** de la firma inglesa **Opus Suplies Ltd**.

La **Discovery 1**, es una unidad integral de disco diseñada especialmente para trabajar con el **ZX-Spectrum**. Sus dimensiones son realmente reducidas (aprox. 21×21×9 cm.), lo que la hace especialmente valiosa a la hora de situarla en nuestro puesto de trabajo, siempre falto de espacio, por grande que sea.

Desde un punto de vista estético la valoración también resulta positiva, tanto por la forma como por el acabado y el color, idéntico al del ordenador.

La parte mecánica está bien resuelta. El ordenador se acopla al **Discovery 1** directamente por medio de su conector posterior y queda apoyado sobre una base metálica solidaria de la carcasa. Ambas piezas se fijan mediante tornillos, lo cual da al conjunto una gran solidez.

No se requiere el alimentador del **Spectrum**, porque la fuente de alimentación incorporada al **Discovery** proporciona energía al conjunto.

Disponemos de cuatro conectores en la carcasa:

- Salida de vídeo para conectar monitor
- Conexión para *joystick*
- Salida **Centronics** para impresora
- Prolongación del conector posterior del **ZX-Spectrum**

Al conectar la **Discovery** al **Spectrum** pueden dejarse conectados los cables TV-VHF y MIC/EAR por si se desea utilizar un TV doméstico y el *cassette* (muy recomendable a efectos de hacer copias de seguridad e intercambio de programas con otras personas). Es una lástima que el constructor no haya previsto los conectores adecuados para facilitar esta maniobra.

La unidad estándar incorpora una unidad de *diskette* **Opus** de 3.5 pulgadas y viene prevista para la incorporación de otra. También está incluido el controlador de disco.

Desde el punto de vista térmico adolece del mismo problema que el **Spectrum**. Aunque puede utilizarse sin problemas en condiciones normales, cuando la temperatura ambiente es algo elevada y se hace uso prolongado, las condiciones dejan algo que desear. El problema es fácil de resolver instalando cualquiera de los ventiladores miniatura existentes en el comercio. También en esta ocasión tenemos que sacar tarjeta amarilla al fabricante.



Todo el Sistema Operativo para el manejo de los discos (DOS) está incluido en una ROM que no ocupa espacio adicional en la memoria RAM. También van incluidas en la ROM las utilidades (copia de discos, compactado de ficheros, etc.) con lo que no es necesario cargar ningún *software* adicional desde disco, como es usual en otros sistemas. En este sentido el manejo resulta realmente cómodo.

El acceso a disco es suficientemente rápido en la práctica para la mayoría de las aplicaciones. En el manual que acompaña el sistema (original de **Opus** y traducción de **Silog**) no figuran datos a este respecto.

Los datos pueden accederse en forma secuencial ([illegible]o en el *cassette*) o de manera aleatoria (directamente a un registro de un fichero). **Discovery 1** incorpora también la facilidad de disco RAM, es decir la posibilidad de utilizar una parte de la RAM como si fuese un disco, con la ventaja de no fatigar el sistema mecánico del disco y aumentar extraordinariamente la velocidad de acceso a los datos.

La elección de *diskettes* de 3.5 pulgadas nos parece adecuada ya que permite un equipo más compacto. Los *diskettes* de 3.5 pulgadas son mucho más robustos y fiables a la hora del manejo que los normales de 5-1/4 pero también considerablemente más caros (aproximadamente el doble). **Discovery** utiliza sólo una cara lo cual entendemos es también una lástima. Con todo disponemos de una capacidad de almacenamiento de 189 Kbytes, distribuidos en configuración de 40 pistas y 18 sectores.

Para aquellos que no conozcan los *diskettes* de 3.5 pulgadas indicaremos brevemente que se diferencian fundamentalmente de los de 5-1/4 pulgadas no sólo en el tamaño. Están compuestos de una funda rígida aunque el disco en sí sigue siendo de material plástico flexible con recubrimiento pelicular magnético. La ranura por donde actúa la cabeza en lectura o escritura

Destino	Observaciones
"d"	Considera el disco como un todo.
"j"	Joysticks.
"K"	Teclado para datos entrantes o parte interior de la pantalla.
"S"	Parte superior de la pantalla.
"m"	Fichero en disco.
"p"	Impresora ZX-Printer.
"t"	Entrada o salida de datos por el port paralelo. Si el codigo del dato a transmitir es inferior a 32 se ignora (excepto para CR,LF) Si es superior a 127 se envia el Token correspondiente y en cualquier otro caso se transmite sin variacion.
"b"	Como "t" pero transmitiendo los datos sin modificar.
"CODE"	Acceso directo a memoria.
"CAT"	Acceso al fichero con todos los datos del directorio del disco.

está permanentemente cubierta por una tapa metálica que se abre automáticamente dentro cuando se requiere leer o escribir datos. De esta manera la protección del *diskette* está bastante más garantizada en cuanto a entrada de polvo o exposición de la película magnética a rayaduras, etc. El sistema de arrastre también resulta más eficaz al disponer de una pieza metálica central solidaria con el disco (en los de 5-1/4 pulgadas no existe parte central y el disco es pillado por un embrague para producir el giro). En lugar de una muesca ocultable mediante una etiqueta adhesiva para dar protección contra escritura, los discos de 3.5 pulgadas disponen de una corredera que puede variarse fácilmente con la punta de un bolígrafo. La posible desventaja de este tipo de discos, a parte de la económica ya mencionada anteriormente, es la de requerirse una mayor precisión mecánica en el posicionado de la cabeza, por ser menor el tamaño de pistas y sectores para una misma capacidad.

En líneas generales el sistema operativo de **Discovery 1** está basado en el **Microdrive**, mejorándole.

El manual suministrado con el equipo está bien presentado, aunque es demasiado conciso. Tampoco hubiera estado mal dedicar algunas páginas a aclarar algunos conceptos de tipo general para ayudar al principiante.

A continuación vamos a aclarar un poco alguno de estos conceptos y a comentar los comandos específicos, aunque sea de una manera esquematizada.

CHANNEL / STREAM (destino, camino)

Con el uso del *software* de **Discovery 1** se puede leer o enviar información a una serie de *channels* que en unas ocasiones son partes físicas como la pantalla o la impresora y otras son ficheros o disco RAM, por ejemplo. No obstante, para poder acceder a estos recursos se necesita un elemento de unión que son los *stream*.

A nuestro modo de ver, el propio término inglés *channel* y las diversas traducciones españolas que circulan por ahí de él (canal, fundamentalmente) generan una gran confusión a los principiantes, sobre todo cuando se utiliza conjuntamente con el término STREAM traducido normalmente por flujo, corriente, etc. Tal vez sea preferible traducir *channel* por destino y *stream* por camino.

Parece más lógico y, sobre todo, más nemotécnico ir (o volver) de un destino (*channel*) a través de un cierto camino (*stream*) que 'ir a un canal por un flujo' o que 'venir de un canal por una corriente'.

El nombre o código dado a los diversos destinos (*channels*) está ligado con el nombre inglés correspondiente y por lo general consta de una sola letra. A continuación se relacionan todos con un breve comentario:

Existen 16 caminos (*streams*) disponibles para que el programador pueda llegar a/de los respectivos destinos (*channel*). No puede llegarse por un camino a dos destinos pero si puede alcanzarse un destino por varios caminos.

Para poder hacer uso de los caminos para acceder a los destinos debe hacerse previamente una asignación mediante el comando OPEN #. Inicialmente el **Spectrum** tiene asignados los caminos 0 y 1 al destino «K», el 2 al «S» y el 3 al «P» pero pueden reasignarse. Por ejemplo si reasignamos el camino 3 al destino «t» cada vez que se ejecute LPRINT o LLIST el resultado saldrá por la impresora paralelo en lugar de por **ZX Printer**.

Los caminos pueden asignarse para transmitir información OUT, o para recibirla, IN. Si no se especifíca nada se toman los valores asignados por defecto en cada caso (ver manual).

Cuando se ha terminado de hacer uso de un camino debe reasignarse o desligarse del destino correspondiente mediante CLOSE # seguido del número correspondiente. De no hacerlo así se corre el riesgo de perder la información que esté aún en el *buffer* asignado al camino y que no haya sido leída o transmitida.

DRIVE (unidad de diskettes)

Discovery admite solamente dos unidades de *diskettes* (bocas). Al de la izquierda se le denomina N°1 y al de

CONCURSO INPUT/ANAYA MULTIMEDIA

¡DESCUBRE AL ASESINO!

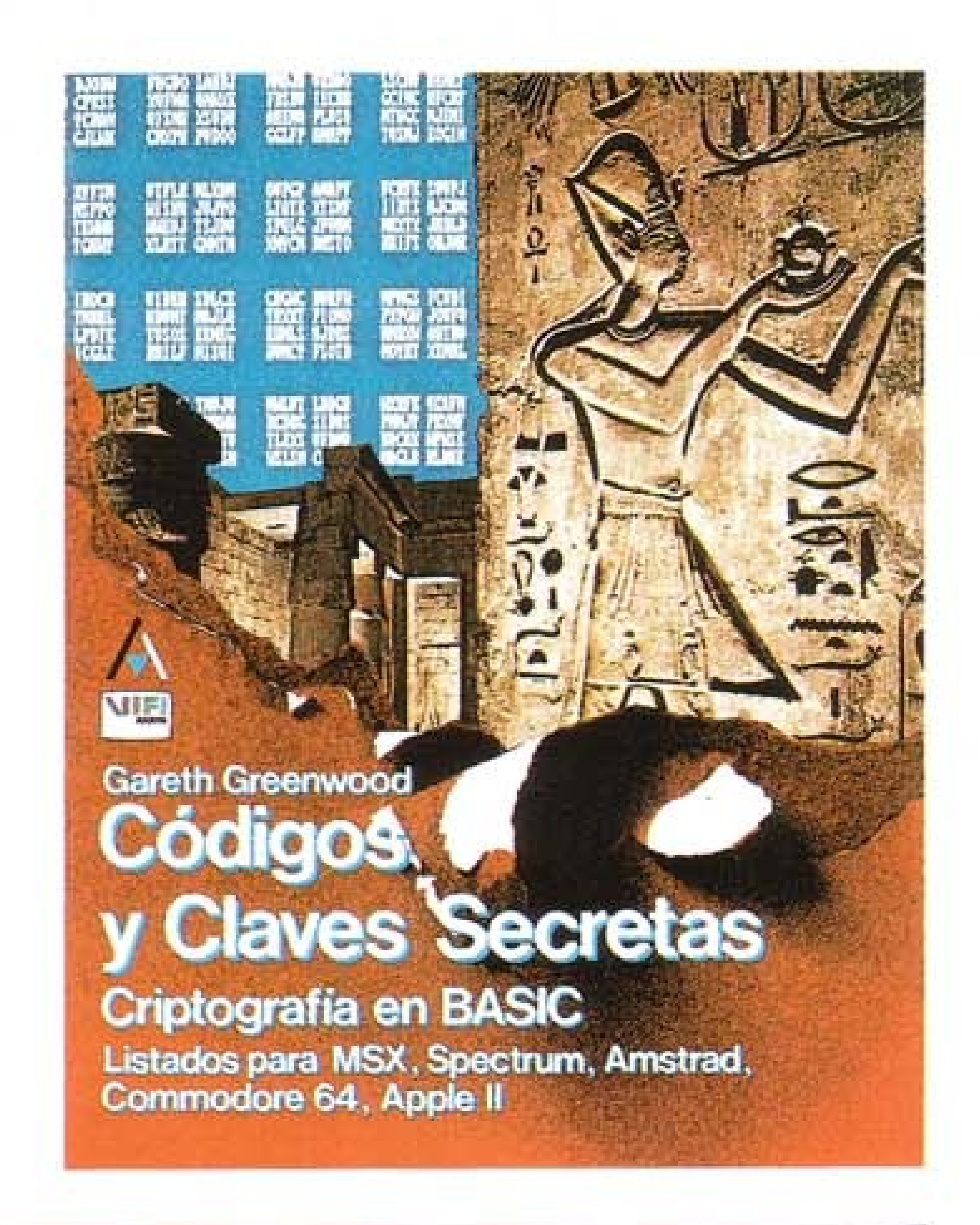
BASES:

Durante tres números sucesivos irán apareciendo en **INPUT** las tres partes del relato «El caso del anciano asesinado» junto con los mensajes cifrados que constituyen las respuestas parciales a la solución del crimen. Cada mes, quienes logréis descifrarlos participaréis en un sorteo consistente en 10 lotes de libros por un valor de 15.000 ptas., a elegir del fondo editorial de **Anaya Multimedia**, y una suscripción gratuita por un año a **INPUT**. Habrá tres sorteos, de tal forma que no será necesario haber descifrado el enigma del mes anterior para optar al premio.

Un fabuloso premio será la guinda final de este pastel. El descubridor de las motivaciones del crimen visitará al próximo **PCW Show** a celebrar en Londres. En caso de haber más de un acertante, recurriremos a la inevitable fórmula de sorteo.

La solución del primer mensaje cifrado deberéis enviarla, junto con vuestros datos personales, a la Redacción de **INPUT** antes del 15 de junio.

Las decisiones del jurado serán inapelables, dándose en las páginas de **INPUT** cumplida cuenta de la marcha del concurso.



Nota: El libro de reciente aparición «Códigos y claves secretas», de ***Anaya,*** *contiene todos los programas útiles para esta labor. Aunque si has trabajado para los servicios de inteligencia de alguna potencia no te será de gran utilidad.*



EL CASO DEL ANCIANO ASESINADO

Repasé la casa y comprobé que el viejo no fumaba, aunque no carecía de algunos vicios cuya enumeración no aportaría nada a este informe.

—El ladrón o los ladrones debieron dejarse un cigarro encendido —comentó mi ayudante.

—Los ladrones no suelen fumar mientras trabajan —respondí.

Di un par de vueltas más, que sólo me sirvieron para comprobar la porquería que es capaz de acumular en una casa un hombre solitario, aun cuando sea tan rico como aparentaba nuestro cadáver maniatado. Revisé de nuevo el interior de la caja fuerte. Había cartas de amor, recortes de periódicos, facturas sin saldar, apremios de pago, títulos de propiedad, una dentadura postiza y una póliza de seguro. Me guardé la póliza y dejé lo demás donde estaba.

En esto, llegó el juez con intención de levantar el cadáver. Tuvo que esperar un poco, porque nuestros expertos no habían terminado de estudiar el modo en el que el viejo había sido maniatado. Al poco, apareció también por el piso un anciano muy parecido al muerto y que dijo ser hermano suyo.

—¿Cómo se ha enterado Vd. de todo esto? —pregunté.

—Vivo en el piso de arriba. Estaba durmiendo la siesta, pero me ha avisado un vecino —respondió llorando frente al espectáculo.

Entonces, mi ayudante me pasó una nota que decía:

```
XNLZNJSYJ HNKWFIT HTQZRSFX HTRUQJYFX XFI
KQ UISIN MFTSN QXZSJ FSFWS TFRSJ JXJYS Z
XFJV FISNF XJJIW JNTTS IXUTF NJRQW YQZTW
 SNTYN JQ
```

Desde la comisaría telefoneé a la casa de seguros de la póliza y me confirmaron lo que ya había leído: que cubría un seguro por robo de una serie de joyas valoradas en quince millones de pesetas, joyas que, según la información de la casa de seguros, permanecían en una caja fuerte de la casa del anciano muerto. Al día siguiente visité a su hermano y le sometí a un complicado interrogatorio, complicado porque el pobre viejo no paraba de llorar.

Juan José Millas



Y a la venta en: EL CORTE INGLES – GALERIAS PRECIADOS – PRYCA – CONTINENTE – HYPERCORD y establecimientos especializados.
MAX HEADROOM
Lo que yo quisiera saber es . . .
Disponible en - CBM 64 SPECTRUM - 48/128 K y AMSTRAD.
Porqué alguien querrá jugar con algo diferente?
QUICKSILVA
Los juegos más poderosos del universo
MIND GAMES ESPAÑA S.A
MARIANO CUBI, 4 ENTLO.
TEL. 218 34 00
08006 BARCELONA
© Chrysalis Visual Programming Ltd.
MAX HEADROOM
Deseo recibir los juegos que a continuación especifico, comprometiéndome al pago del importe de los mismos.
Nombre
Dirección
Teléfono
Firma:
SISTEMA
CANTIDAD
Contrarreembolso.
Adjunto Talón.
Giro Postal.
Deseo recibir información de sus programas en:
MSX
AMSTRAD

la derecha Nº2. No obstante a efectos operativos reconoce el Nº3 y el Nº4 para distinguir entre *diskette* fuente y *diskette* a copiar (3 izq., 4 dcha.) en el caso de copias de *diskettes* haciendo sólo uso de un sistema. El Nº5 lo reserva para el disco RAM.

Transmisión global de información entre destinos

Existe la posibilidad de transferir en bloque información de un destino a otro, como por ejemplo un fichero de disco a la pantalla, o a otro fichero. Otra cosa es que tenga sentido o aplicación práctica en todas las ocasiones, ya que la información se transmite tal cual está almacenada internamente. Ver más adelante comando MOVE.

FILES (Ficheros)

Los ficheros pueden ser de programas, pantallas (SCREEN$), bytes de memoria (CODE), o matrices (DATA) y deben estar identificados por un nombre (1 a 10 caracteres válidos).

Ficheros de datos

Los ficheros de datos son unidades lógicas de inforamción compuestas de registros, los cuales contienen una serie de bytes que pueden agruparse en campos (aún cuando no se dispone de acceso individualizado).

Antes de poder introducir datos en un fichero es necesario crearlo; es decir reservar espacio suficiente en disco.

Para tener acceso a un fichero ha de asignársele un camino mediante OPEN # <camino>;<destino>;<unidad>;<nombre><tipo>. Con tipo definimos si se va a escribir o leer.

La introducción de registros en un fichero se realiza mediante PRINT # <camino>;<variable>, mientras que para la lectura se utiliza INPUT # <camino>;<variable>. Si se desea leer sólo un caracter utilizaremos INKEY$ <canal> a través de una variable.

Anter de continuar conviene hacer una aclaración para aquellos que manejen discos por primera vez. Existen dos maneras de escribir y leer los datos de un fichero: secuencialmente (uno a continuación de otro) o en forma aleatoria (acceso en el orden que se desee). Con **Discovery 1** se puede operar de las dos maneras.

Acceso secuencial

Al tratar de asignar un fichero mediante OPEN # <camino>;<destino>;<nombre> éste queda asignado para leer, si es que ya existe en el disco. Si no es así, el ordenador creará uno y el fichero quedará automáticamente en modo escritura.

Hay que tener en cuenta que si la asignación se hace mediante OPEN # <camino>;<destino>;<unidad> OUT también se creará un fichero y quedaremos en modo escritura, pero si hubiese otro fichero con el mismo nombre, éste desaparecería.

Si lo que se pretende es añadir registros a un fichero existente ésto debe indicarse al efectuar la asignación de camino añadiendo a OUT el número máximo de bytes que deseamos tenga el fichero o sustituyendo OUT por EXP seguido del número de caracteres con que queramos ampliar el fichero. Es decir, de alguna manera hemos tenido que ampliar su capacidad.

Si no se especifica el número de caracteres de la ampliación o se indica '−1' ésta quedará fijada automáticamente en la mitad del espacio disponible, y si esto no es posible, hasta la frontera con el siguiente fichero en el disco.

Cada registro puede tener cualquier longitud y para tener acceso en lectura a cualquiera de ellos ha de pasarse necesariamente por todos los anteriores. A la hora de la escritura siempre se escribe 'el siguiente'.

Acceso aleatorio

Un fichero de acceso aleatorio o directo debe tener una estructura fija y bien definida. Inicialmente debe abrirse el fichero con OPEN # <camino>;<destino>;<unidad>RND <número máximo de caracteres por registro>,<número de registros>. Si el último valor se fija en '−1', el ordenador elegirá el número de registros que quepan en la mitad del espacio disponible en disco hasta el fichero siguiente.

Para acceder a un fichero que ya exista debe asignarse un camino, común a lectura/escritura, con OPEN # <camino>;<destino>;<unidad> RND <longitud de registro>. Se utilizará PRINT#, INPUT# o INKEY$# como anteriormente.

Lo verdaderamente valioso es la posibilidad de poder fijar el puntero de lectura o escritura antes de efectuar éstas por medio de POINT # <camino>;<número de registro>. Es decir, se puede leer directamente el contenido del registro 23 o alterar el del número 2, por ejemplo.

No debe olvidarse nunca cerrar un fichero (CLOSE#<camino>) al finalizar su manejo.

Compactado de ficheros

Inicialmente los ficheros se almacenan en el disco en el orden en que se generan, uno a continuación del otro, sin dejar espacios intermedios.

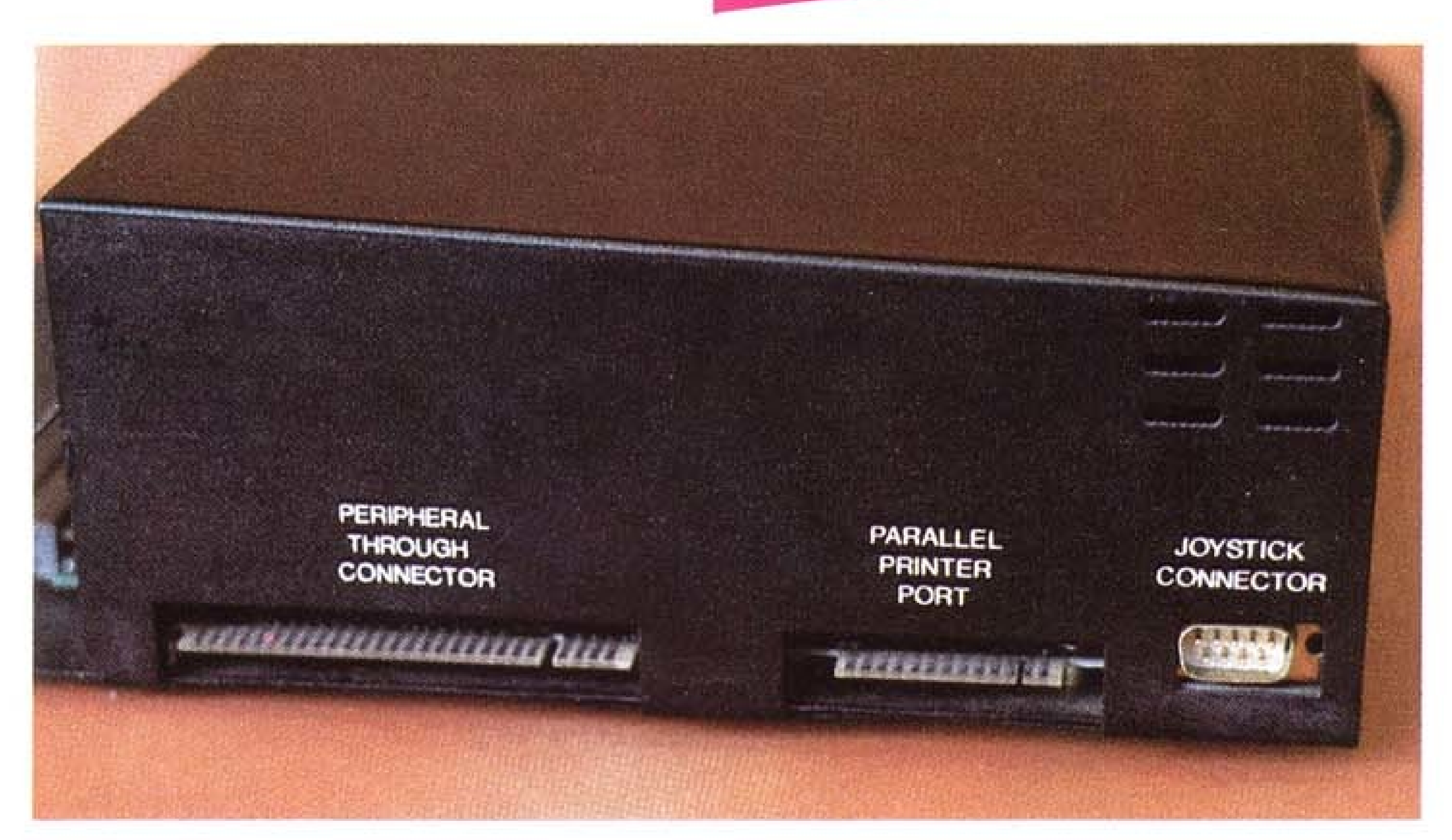
A medida que se van generando, copiando, anulando o modificando ficheros se van originando espacios muertos entre ellos de forma que aún existiendo espacio vacío suficiente en el disco llegue un momento que no podamos almacenar más ficheros. Mediante la operación de 'compactación' o 'reagrupación' podemos agrupar todos los espacios intermedios y reutilizarlos.

Para ello basta utilizar el comando MOVE «d»;1 TO «d»;1 (en el caso de la unidad de *diskette* izquierda).

Catálogo

El índice o catálogo de toda la información archivada en el disco no es precisamente uno de los puntos fuertes de **Discovery 1**. A petición del comando CAT#<camino>;<unidad> (el valor por defecto es el camino 2, es decir, para sacar el catálogo del disco izquierdo por pantalla basta CAT 1) aparece la lista de «cosas» que hay en disco, pero no podemos distinguir si se trata de un fichero de datos, un programa BASIC, una pantalla, etc.

Mediante la manipulación un tanto compleja (ver pág. 28 del manual inglés) de una copia del fichero del CAT (mejor no manipular el original, por riesgo de perder toda la información del disco) podrás conocer el número



de bytes en el último bloque del archivo −1, el bloque de comienzo y el bloque final.

Otro inconveniente, aunque un tanto relativo, es el de no poder repetir nombres aunque se trate de ficheros de distinto tipo (por ej. mismo nombre a un fichero de código máquina que a un programa BASIC).

Autoarranque

Discovery 1 ofrece una opción interesante que consiste en que mediante la sola pulsación de las teclas RUN y ENTER se pone en marcha un programa denominado 'run' en el disco número 1. Esta opción podemos aprovecharla para un directorio comentado, páginas de ayuda, etc.

Información sobre errores

En el manual aparece una lista con trece informes ante situación de error que no requiere mayor explicación. Tampoco estaría mal el que el fabricante hubiera facilitado la rutina para poder manejarlos.

Carga, archivo y verificación

Estas tres funciones son análogas a las de *cassette*, con la única diferencia que hay que añadir *<destino>;<unidad> entre LOAD, SAVE o VERIFY y los datos que pondríamos en *cassette*.

Comandos

Resumimos a continuación los comandos adicionales que nos ofrece **Discovery 1**, sin comentarios:

CAT # <camino>;<unidad>
CLEAR # <camino>
CLOSE # <camino>
CLS #
CODE <dir.comienzo>,<núm. de bytes>
DATA <nombre>()
ERASE <archivo>
FORMAT «j»;<estado>
FORMAT «m»;<unidad>;<nombre>
INKEY$ # <camino>
INPUT # <camino>;variable1,variable2,...
LINE <número>
LOAD * < especif. destino>
LPRINT # <camino>;variable1,variable2...
LLIST #
MERGE * < especif. destino>
MOVE <destino> TO <destino>
OPEN # <camino>;<especif. destino><inf aux>
POINT # <camino>;<registro>
PRINT # <camino>;variable1,variable2...
SAVE * <especif. destino>
VERIFY * <especif. destino>

Ejemplo de programación

Para aquellos que se aventuran por primera vez en el mundo del *diskette* indicamos un caso práctico, que aunque elemental, de la parte que consideramos más interesante.

```
10 REM**PROGRAMA DEMO
   DISCOVERY 1
20 LET UNIDAD=1
30 CLEAR #:DIM N$(9):OPEN #4
   ;"m";UNIDAD;"DEMO1" RND
   10,10
40 FOR I=1 TO 10: INPUT
   "NOMBRE "+STR$ I+" ? ";N$
   :PRINT #4;N$:NEXT I
50 CLOSE #4
60 GO SUB 600
90 INPUT " MODIFICAR ? S/N "
   ;P$: IF P$<> "S" AND P$ <
   > "s" THEN STOP
100 OPEN #4;"M";UNIDAD; RND
    20
110 INPUT " NUM. DE REGISTRO
    ? ";N:IF N<1 OR N>10
    THEN GO TO 110
130 POINT #4;N: INPUT #4;N$
    :PRINT "NIMBRE ACTUAL
    DE ";N;TAB 22;N$
140 INPUT "NUEVO NOMBRE ? ";
    N$:POINT #4;N: PRINT #4;
    N$
150 CLOSE #4: GO TO 60
300 REM**USO DEL DISCO RAM
310 CLEAR 32000: LET UNIDAD=
    5: FORMAT "m";UNIDAD;
    "DEMO1"
320 GO TO 30
600 REM**LISTADO FICHERO
    DEMO1
610 OPEN #6;"m";UNIDAD;
    "DEMO1";RND 20
620 FOR I= 1 TO 10: INPUT #6
    ;N$:PRINT I;TAB 5; N$:
    NEXT I
630 CLOSE #6: RETURN
```

Se trata de generar un fichero en disco con 10 nombres, comprobar que se han grabado estos 10 datos y luego modificarlos a voluntad.

Comenzando por la instrucción 300 se hace lo mismo, pero desde un 'disco RAM '. Puede observarse la gran diferencia de agilidad de manejo. La diferencia sería notabilisima para un fichero de grandes dimensiones.

Como final de todo podemos decir que **Discovery 1**, a pesar de algunas de sus limitaciones ofrece unas posibilidades muy interesantes a todo aquel que desea hacer algo profesional. La lástima es que, salvo ignorancia del firmante, no hay software disponible en cantidad apreciable.

MODELIZANDO LA REALIDAD

Elegir la combinación ganadora en un juego de azar es equivalente a la extracción de números aleatorios. Es una tarea adecuada para tu micro, siempre que poseas unas nociones de estadística.

Los ingenieros de todas las especialidades construyen modelos, estructuras en miniatura, para probar sus diseños. Al hacer esto están simulando un sistema real. En el artículo de simulación, que presentamos hace unos meses, vimos que tu micro puede hacer algo semejante, como es ir recorriendo las etapas sucesivas de un determinado proceso para dar un resul tado que se parezca a algo que ocurre realmente. Aquí te presentamos la manera de combinar los principios de la estadística y de la modelización para resolver problemas de una forma realista, tanto en juegos como en situaciones de la vida real.

TIPOS DE MODELOS

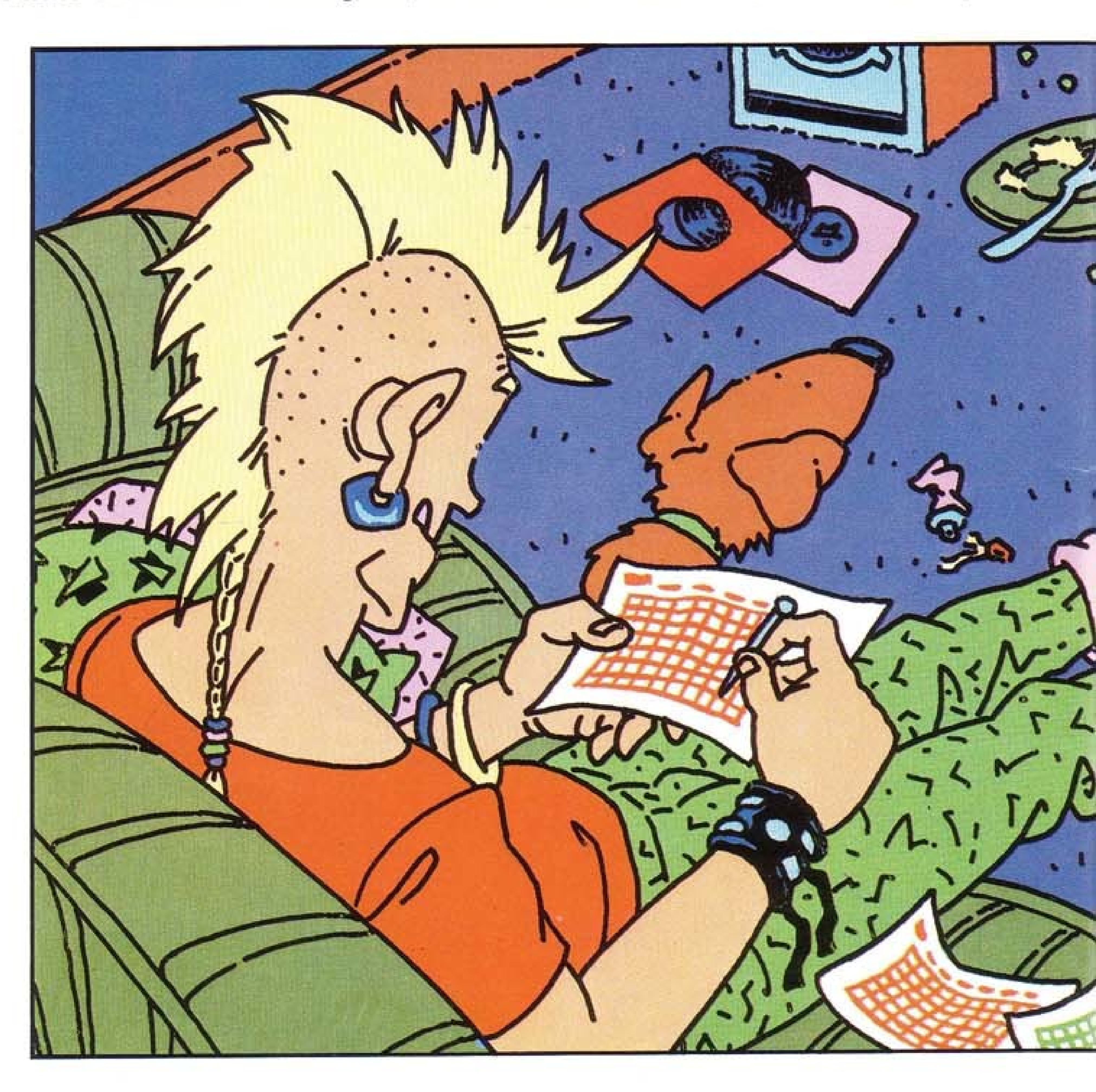
Técnicamente un modelo es una representación de un sistema real, que se construye por su sencillez y utilidad como ayuda en la toma de decisiones. Hay tres categorías de modelos: iconográficos, analógicos y simbólicos, aunque no todos resultan practicables en la simulación por parte de los entusiastas del micro doméstico.

Los modelos iconográficos no tienen partes móviles. A esta categoría pertenecen los mapas de carreteras y los aviones de juguete. Por ejemplo, en los estudios previos para la construcción de un puente sobre un estuario, los ingenieros de caminos podrían construir un modelo del puente y de la zona que rodea al estuario, para estudiar los efectos de la marea, las riadas y el viento sobre la estructura. Aunque se emplean ordenadores para monitorizar estos factores y analizar los resultados, el hecho de tener que recopilar datos a partir de ensayos reales significa que no hay necesidad de realizar una simulación.

En los modelos analógicos se representa una cantidad por otra. Por ejemplo el profesor **Phillips**, un economista británico, dispuso unos cuantos depósitos llenos de agua fría a diferentes alturas y utilizó el flujo de líquido entre los tanques para representar el flujo monetario en la economía del Reino Unido. Puede hacerse lo mismo electrónicamente, representando las diferentes cantidades por medio de diferentes valores de tensión eléctrica, pero para esto se emplean generalmente ordenadores analógicos, en lugar de los digitales, como es un micro doméstico. La tercera categoría, el modelo simbólico o abstracto, es la que resulta especialmente interesante para el que quiera llevar a cabo simulaciones en un ordenador doméstico.

INCERTIDUMBRES DE LA VIDA REAL

En los modelos de tipo simbólico, se utiliza una fórmula matemática para representar el sistema real. Por ejemplo, si quieres estudiar la distancia (S) recorrida por un coche que viaja a velocidad constante (V) durante un cierto tiempo (T), el modelo adecuado sería S = V*T. Este tipo de



ecuación se llama determinística, debido a que no da ninguna medida de las incertidumbres. Describe con precisión el movimiento del automóvil para cualquier conjunto de valores que especifiques. Sin embargo existen muchos sucesos que no pueden ser especificados sin un cierto grado de incertidumbre. Tienes un ejemplo de esto en el lanzamiento al aire de una moneda. Los modelos que describen este tipo de sucesos imprecisos se llaman estocásticos. Normalmente son este tipo de sucesos los que requieren simulación, particularmente en los juegos.

Algunas variables pueden tomar solamente valores discretos, tales como 0, 1, 2, 3, ... A esta categoría pertenecen el número de goles en un partido de fútbol o el número de hijos varones de una familia. Cuando se simulan variables aleatorias de este tipo, es importante darse cuenta de que existen muchos procesos aparentemente diferentes que tienen de hecho la misma estructura subyacente.

Consideremos por ejemplo el caso de la simulación del lanzamiento de una moneda al aire, anteriormente mencionado. Este programa podría modificarse fácilmente para representar los resultados aleatorios de una partida de dados o el número de goles conseguidos en una serie de lanzamientos de *penalties*. En todos estos casos hay un cierto número de pruebas repetidas independientes, cuyos resultados pueden producirse con probabilidades bien definidas. Supongamos que la prueba consiste en lanzar una moneda al aire cuatro veces. El resultado de cada prueba puede ser cuatro caras, tres caras y una cruz, dos caras y dos cruces, una cara y tres cruces o cuatro cruces. Además hay una probabilidad bien definida asociada con cada uno de estos posibles resultados.



Las estructuras de este tipo, llamadas procesos de **Bernoulli**, generan variables aleatorias discretas que no resultan adecuadas, por ejemplo, para el estudio del número de accidentes que ocurren en un **Grand Prix**, el número total de llamadas telefónicas que se producen en un período de media hora o el número de rayos registrado durante una tormenta eléctrica. En todos estos ejemplos, los sucesos se producen de modo casual durante un determinado período de tiempo, ellos solos y no como consecuencia de un conjunto de pruebas repetidas. A estas estructuras se les llama procesos de **Poisson**.

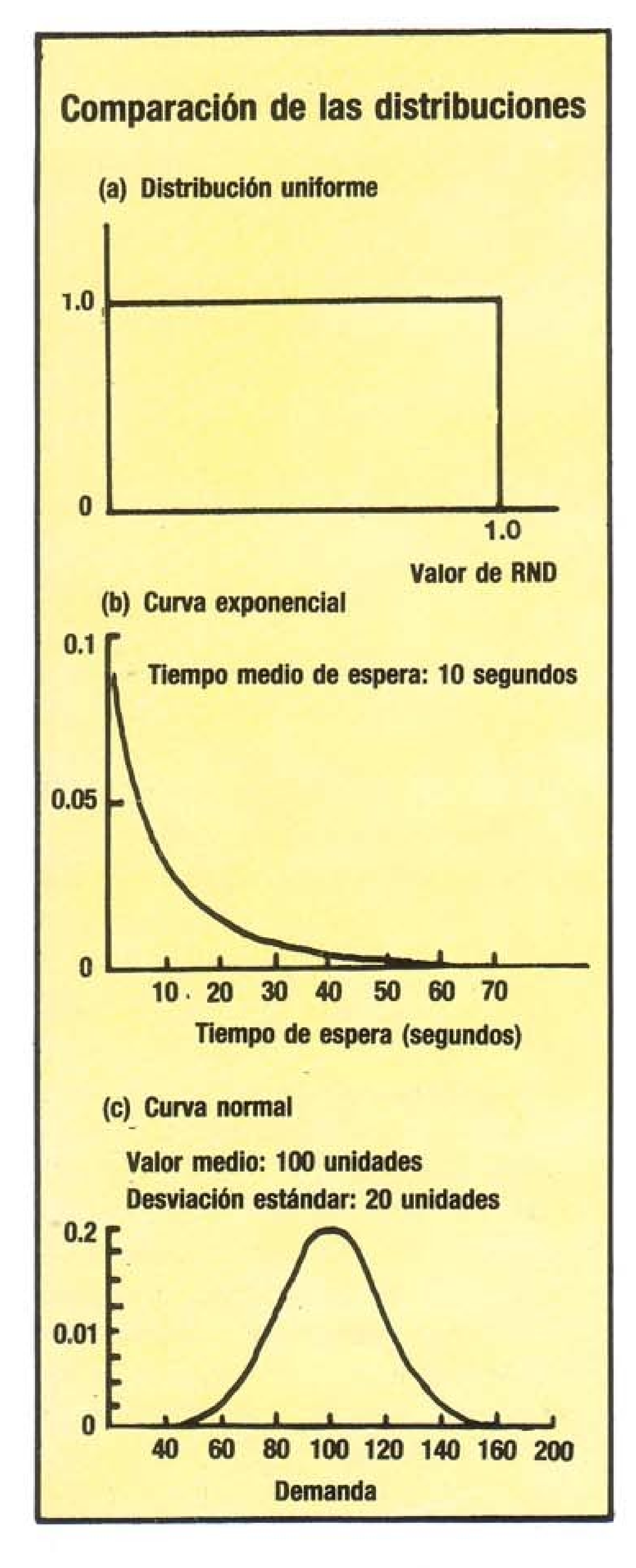
Teclea el siguiente programa que te permite realizar un muestreo a partir de cualquier distribución de **Poisson** que elijas:

```
10 BORDER 0: PAPER 0: INK 7:
   CLS
20 POKE 23658,0
40 PRINT AT 1,6; INVERSE 1;
   " SIMULACION DE POISSON "
   : PRINT: PRINT
50 INPUT "CUAL ES EL VALOR
   MEDIO?";A
70 INPUT "NUMERO DE
   MUESTRAS ";n
80 FOR i=1 TO n
90 LET c=0: LET t=1
100 LET s=EXP (-a)
110 LET t=t*(RND*1)
120 IF t<=s THEN PRINT ;"[5
    *ESPACIO]";c,: GO TO 150
130 LET c=c+1
140 GO TO 110
150 NEXT i
160 INPUT " OTRAS MUESTRAS
    (s/n)?";g$
170 IF g$="s" THEN GO TO 10
180 STOP
```

Al ejecutar el programa, se te pide que introduzcas un valor promedio (la media aritmética) y el tamaño de tu muestra (líneas 50 y 70). La parte principal del programa (líneas 80 a 150) utiliza una ecuación para generar las variables de **Poisson**. A la variable S se le asigna (línea 100) el valor del número e (una constante matemática) elevado a la potencia A (el valor de la media que tú has introducido). La línea 110 elige un número aleatorio entre 0 y el valor de T, y realiza un cambio de escala según el valor de T. La línea 120 compara S y T para decidir si hay que presentar o no una variable (C).

Supongamos por ejemplo que estás diseñando un juego espacial en el que una nave interplanetaria tiene que atravesar una tormenta de meteoritos. Si el número medio de «impactos» en cualquier período de un minuto es dos y se requieren cinco simulaciones separadas de un minuto cada una, el ordenador podría presentar los números 2, 3, 2, 1, 0 como estadística de impactos. Cada elemento de la lista no será muy diferente de 2 (la media), y en total hay cinco elementos (el tamaño de la muestra).

¿Podría utilizarse esta misma técnica para simular, digamos, el tanteo final de un partido de fútbol? Está claro que se trata de una variable discreta, ya que el número de goles conseguidos sólo puede tomar valores enteros: 0, 1, 2, 3, etc. Además son sucesos que se producen por sí solos en el tiempo y no como resultado de pruebas repetidas. En consecuencia se trata de un caso susceptible de ser adecuadamente descrito mediante un proceso de **Poisson**. Aparte del número de partidos jugados, la única información complementaria que necesitas en



este caso es qué valor hay que utilizar como número medio de goles. En este punto se hace necesario salir al mundo real y recoger algunos datos.

Tomando como base los goles marcados en y fuera de casa durante la Liga una elección razonable para los tanteos sería adoptar el valor 1.7 como media de goles domésticos y 1.0 como media de goles fuera de casa. La división por dos de cada uno de estos valores conduce a un tanteo medio por cada medio partido de 0.85 y 0.5. La construcción del programa resulta ya inmediata:

```
20 DIM a$(25,10): DIM h$(25
   ,10): DIM h(50): DIM a
   (50): DIM f(50): DIM g(50)
30 BORDER 0: PAPER 0: INK 7:
   CLS
50 PRINT AT 0,8; INVERSE 1;
   "RESULTADO FINAL "
70 INPUT " CUANTOS PARTIDOS
   (1-25)? ";n
75 IF n<1 OR n>25 THEN GO TO
   70
80 FOR i=1 TO n
90 PRINT "PARTIDO ",i
100 INPUT "EQUIPO DE CASA";
    h$(1)
110 INPUT "EQUIPO VISITANTE"
    ;a$(1)
120 NEXT i
130 PAUSE 400
140 PRINT
150 FOR i=1 TO 2*n
160 LET c=0: LET t=1
170 LET s=EXP (-.85)
180 LET t=t*(RND*1)
190 IF t<=s THEN LET h(i)=c:
    GO TO 220
200 LET c=c+1
210 GO TO 180
220 NEXT i
230 FOR i=1 TO 2*n
240 LET c=0: LET t=1
250 LET s=EXP (-.5)
260 LET t=t*(RND*1)
270 IF t<=s THEN LET a(i)=c:
    GO TO 300
280 LET c=c+1
290 GO TO 260
300 NEXT i
310 CLS
320 PRINT TAB (7);"RESULTADO
    INTERMEDIO";''
330 FOR i=1 TO n
340 PRINT h$(i);h(i);TAB 20;
    a$(i);a(i): PRINT
350 NEXT i
360 PRINT "PULSA UNA TECLA
    PARA CONTINUAR"
370 IF INKEY$="" THEN GO TO
    370
375 CLS
380 PRINT TAB 12;"RESULTADO
    FINAL"''
390 FOR i=1 TO n
400 LET f(i)=h(i)+h(n+i)
410 LET g(i)=a(i)+a(i+n)
420 PRINT h$(i);f(i);TAB 20;
    a$(i);g(i): PRINT
430 NEXT i
440 INPUT "OTRA VEZ (S/N)? "
    ;g$
450 IF g$="n" THEN GO TO
    490
460 INPUT " LOS MISMOS
    EQUIPOS (S/N)? ";p$
470 IF p$="n" THEN GO TO
    70
480 GO TO 150
490 STOP
```

Ejecuta el programa y responde a las preguntas que te hace su primera sección (desde el principio hasta la línea 120). Para evitar el tener que teclear demasiado, puedes introducir letras del alfabeto en vez de nombres reales para ambos equipos. Las líneas 150 a 300 utilizan el algoritmo de **Pois-**



son *que vimos en el primer programa*, para generar los tanteos del medio tiempo en casa y fuera de casa. Las líneas 320 a 350 organizan la presentación de las puntuaciones de medio tiempo y las líneas 390 a 430 se ocupan del resultado de los partidos completos.

Aunque los goles individuales simulados por partido no correspondan con los resultados reales, el comportamiento global de la situación simulada debería parecerse bastante a dichos resultados reales.

La distribución de **Poisson** funciona bien con variables que tomen valores enteros; pero hay otras variables aleatorias continuas que toman valores fraccionarios. Por ejemplo en una simulación de ordenador del salto de longitud en una **Olimpiada**, un participante podría saltar una distancia cualquiera entre seis y nueve metros. La longitud de dicho salto es un ejemplo del tipo de variable aleatoria continua.

Aunque la función RND del ordenador es adecuada para la simulación de variables aleatorias discretas de tipo **Poisson**, necesitas una función RND modificada para la modelización con variables continuas. La figura 1 compara el tipo de distribución dado por ambos tipos de variables. Las variables discretas dan una distribución uniforme, mientras que las variables continuas pueden dar una distribución exponencial o normal.

Teclea el siguiente programa que le permitirá transformar las variables aleatorias planas o uniformemente distribuídas que te da la función RND de tu ordenador en variables aleatorias con distribución exponencial.

```
10 BORDER 0: INK 7: PAPER 0:
   CLS
20 POKE 23658,0
40 PRINT AT 0,4; INVERSE 1;
   " SIMULACION EXPONENCIAL
   "''
50 INPUT "CUAL ES EL VALOR
   MEDIO? ";a
70 INPUT " MEDIDA DE LA
   MUESTRA? ";n
80 FOR i=1 TO n
90 LET x=(-a)*LN (RND*1):
   LET y=INT (x*10): PRINT
   "[6*ESPACIO]";1*y,
100 NEXT i
110 INPUT " OTRA SIMULACION?
    (s/n)";g$
120 IF g$="s" THEN GO TO 30
130 STOP
```

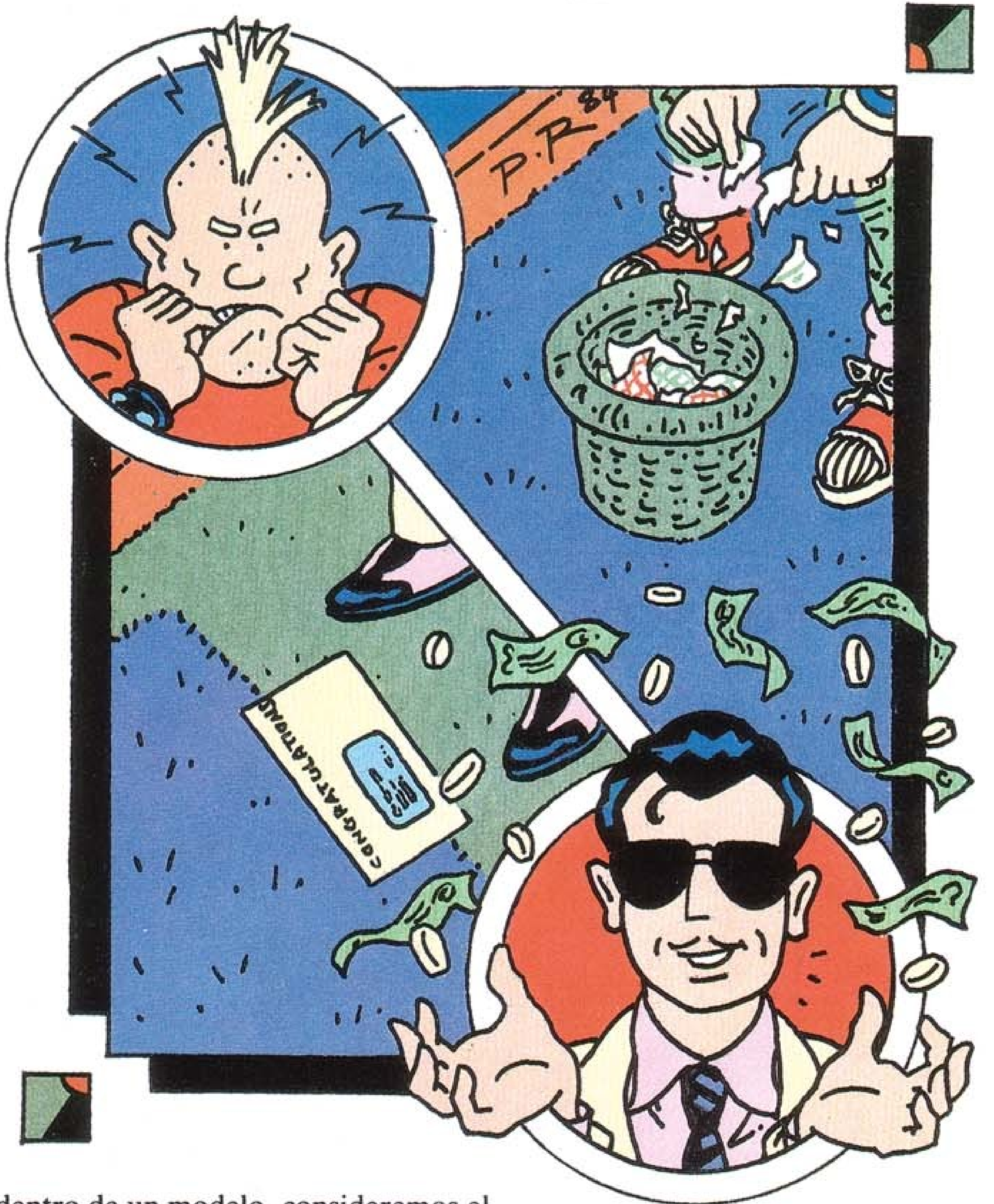


Al ejecutar este programa tienes que especificar el valor medio y el número de muestras requeridas; a continuación aparecerán en pantalla las variables. Todo se hace en la línea 90. Se utiliza una técnica matemática llamada método de la transformada inversa, para convertir variables «planas» en variables exponenciales. Este programa puede usarse también perfectamente para simular, por ejemplo, el tiempo de espera de una secadora en una lavandería, o la vida media de los paquetes de energía en un juego de aventura espacial. En todos los casos el concepto importante es el del tiempo transcurrido.

DISTRIBUCION NORMAL

La curva normal en forma de campana representa la más famosa de las distribuciones estadísticas. Ha resultado muy útil en la descripción de todo tipo de fenómenos naturales, incluyendo las tallas y los pesos de las personas adultas. Además está ampliamente aceptada como el medio más apropiado de describir la distribución de errores.

Para ilustrar la forma en que podría incorporarse la distribución normal



dentro de un modelo, consideremos el siguiente ejemplo para diseñar un juego comercial. Supongamos que las ventas (S) de una empresa dependen de algún nivel base (S0) y de la cantidad que se ha gastado en publicidad (A). Esta relación podría resumirse mediante le expresión S = S0 + (B*A), siendo B una constante conocida.

Pero en la vida real raramente se dan estas relaciones tan precisas. Los sucesos y circunstancias imprevistos distorsionan los hechos o introducen «ruido» en el sistema. Este sistema se puede modelizar adecuadamente introduciendo una ecuación de perturbación con distribución normal para las ventas. La ecuación completa para la cifra de ventas sería entonces S = S0 + (B*A) + Distribución normal.

Para ver cómo funciona, teclea el siguiente programa:

```
10 BORDER 0: PAPER 0: INK 7:
   CLS
40 DIM u(50): DIM y(50): DIM
   z(50)
50 PRINT AT 0,7; INVERSE 1;
   " SIMULACION NORMAL "''
60 INPUT " CUAL ES EL VALOR
   MEDIO? ";a
70 INPUT " CUAL ES EL VALOR
   MINIMO? ";m
80 INPUT " NUMERO DE
   MUESTRAS? ";n
90 LET b=(a-m)/2.5
100 FOR i=1 TO n
110 LET t=0
120 FOR j=1 TO 15
130 LET y(j)=RND*1
140 LET t=t+y(j): NEXT j
150 LET u(i)=((t/15)-.5)
   /SQR
   (1/(12*15))
160 LET z(i)=a+(b*u(i))
170 LET p=INT (10*z(i)):
    PRINT "[5*ESPACIO]";.1*p,
180 NEXT i
190 INPUT " OTRA MUESTRA? "
    ;g$
200 IF g$="y" THEN GO TO 50
210 STOP
```

Ejecuta la simulación y especifica los valores medio y mínimo de tu variable. Técnicamente, el margen de valores de la curva normal no está acotado, por lo que existe una pequeña probabilidad de que un valor simulado sea menor que el valor mínimo especificado por tí. Pero esto no sucederá muy a menudo.

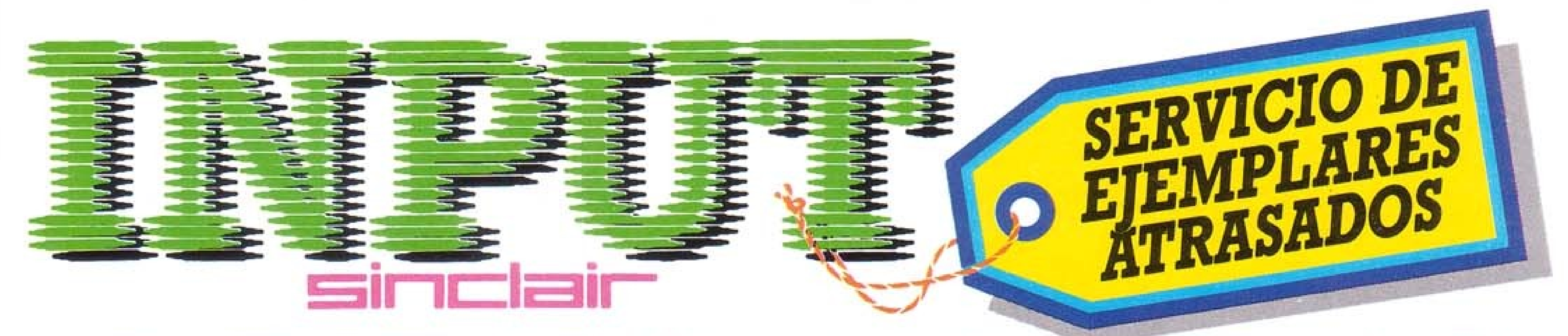
Las líneas 120 a 150 utilizan la función RND para generar 15 números aleatorios entre 0 y 1 y a continuación sumarlos. La línea 160 calcula la media y le añade un factor de escala. Después de otro cambio de escala (línea 160) se puede imprimir el resultado.

Tal vez te ayude a tener una sensación más real de lo que ocurre, el hecho de hacer que los números se generen a partir de dos distribuciones con la misma media y aproximadamente el mismo margen de valores, pero con diferentes formas. Ejecuta este programa introduciendo los valores de 100 para la media, 50 para el valor mínimo y 40 para el tamaño de la muestra. Con los valores obtenidos puedes hacer un gráfico parecido al de la figura 1C.

Borra ahora la línea 90 y las líneas 120 a 160, y escribe de nuevo la siguiente línea:

```
110 Z(I)=M+RND(1)*2*(A-M)
```

Estos cambios hacen que el programa sea capaz de presentar valores aleatorios con distribución uniforme. Ejecuta de nuevo el programa, introduciendo los mismos valores que antes y compara los resultados. Al dibujar esta vez los valores se obtiene un gráfico como el de la figura 1B. Observa que las lecturas normales (las de la primera prueba) están mucho más apiñadas en torno al valor medio.

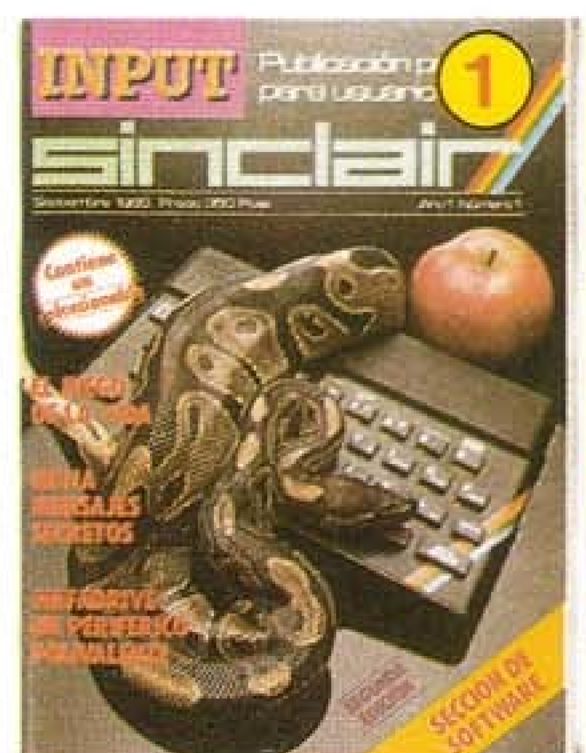


¡NO TE PIERDAS NI UN SOLO EJEMPLAR!

INPUT SINCLAIR quiere proporcionar a sus lectores este nuevo servicio de ejemplares atrasados para que no pierdan la oportunidad de tener en sus hogares todos los ejemplares de esta revista, líder en el mercado español.

Podréis solicitar cualquier número de **INPUT SINCLAIR** que querais, siempre al precio de cubierta (sin más gastos).

Utiliza el cupón adjunto, enviándolo a **EDISA** (Dpto. de Suscripciones), López de Hoyos, 141 - 28002 Madrid, o bien llámanos por teléfono al (91) 415 97 12.

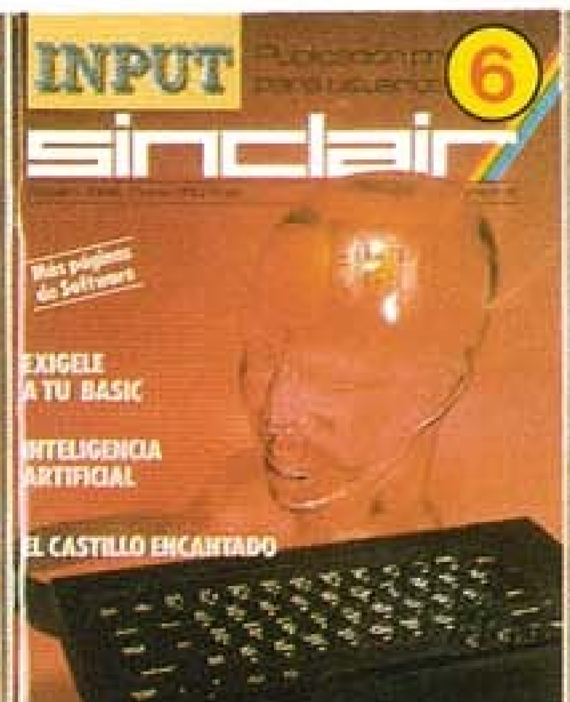


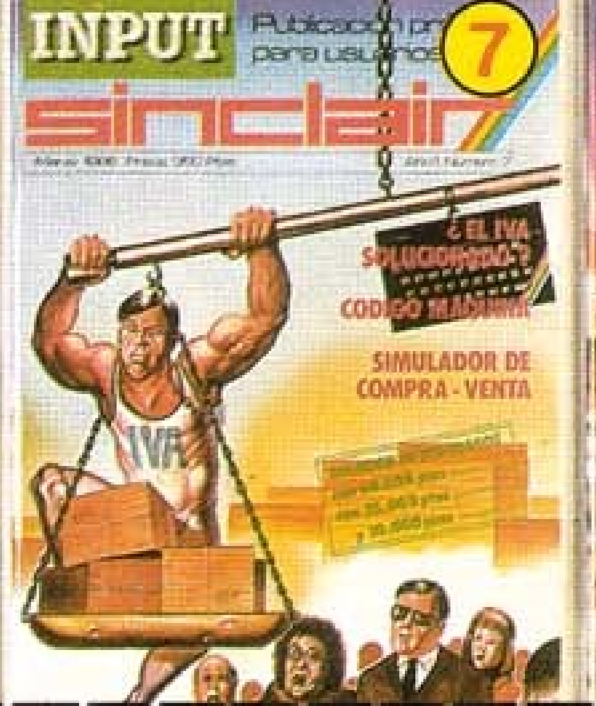


















CUPON DE PEDIDO

SI, envíenme contrarreembolso .... ejemplares de **INPUT SINCLAIR** de los números:

(marca con una (X) tu elección)

[1] [2] [3] [4] [5] [6] [7] [8] [9]

NOMBRE

APELLIDOS

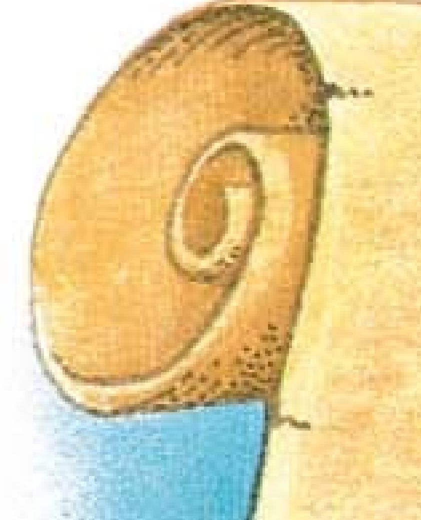
DOMICILIO

NUM. PISO ESCALERA COD. POSTAL

POBLACION PROV.

TELEFONO FIRMA





LOS MEJORES DE INPUT SINCLAIR

PUESTO	TITULO	PORCENTAJE
1.°	Commando	18,4 %
2.°	Rambo	18,1 %
3.°	Saboteur	14,7 %
4.°	Winter Games	9,8 %
5.°	Three Weeks in the Paradise	8,1 %
6.°	Gunfright	8,1 %
7.°	Ole Toro	6,5 %
8.°	Camelot Warriors	6,5 %
9.°	Bady Works	4,9 %
10.°	Robin of Wood	4,9 %
		100 %

Para la confección de esta relación únicamente se han tenido en cuenta las votaciones enviadas por nuestros lectores de acuerdo con la sección «Los Mejores de Input».

Junio de 1986



EL ATAQUE DE LOS «BOINAS VERDES»

Parece que las casas de software han encontrado un nuevo «filón» en los programas de tema bélico. Prueba de ello son los ya famosos **«Rambo»** y **«Comando».** Pero por si todavía no tuviéramos bastante, **Konami** acaba de lanzar al mercado un programa que estamos seguros será un éxito: **Green Beret** (boina verde).

Green Beret es una novedad de tan reciente aparición que todavía nos quema en las manos. Sin embargo, el «boina verde» es un viejo conocido de los aficionados a las «maquinitas» de los bares y las salas de juegos. Por eso estamos seguros de que muchos lectores se saltarán la larga parrafada colocada bajo este epígrafe. Quedan disculpados.

Green Beret es un juego trepidante. Desde el primer segundo de juego hasta el último, te encontrarás completamente solo, luchando contra cientos de perversos soldados, soviéticos, por supuesto, que tratarán de aniquilarte por todos los medios a su alcance. Y tales medios no son pocos: les basta con tocarte, aunque también pueden pegarte una patada de kárate, o un certero disparo, o lanzarte una granada de mortero, o colocar una mina justo, justo por odnde tú vas a pasar. En las instrucciones del juego dicen que también muerden, pero aunque lo hemos probado a fondo, todavía no lo hemos visto. Si es una metáfora, está muy bien puesta.

Ante tal despliegue de medios en tu contra, lo único que puedes hacer por sobrevivir es saltar, subir y bajar por escaleras y plataformas, tumbarte, y no estarte quieto ni un segundo. Tu única arma es un mortífero cuchillo que podrás manejar diestramente con sólo apretar un botón, pero también puedes capturar armas al enemigo y usarlas contra él. Estas pueden ser de tres tipos: lanzacohetes, lanzallamas, y granadas. El único inconveniente que tienen es que se agotan rápidamente, por lo que deberás restringir su uso siempre que sea posible.

El juego se basa en un desarrollo de pantalla «en banda». Esto quiere decir que vas avanzando siempre hacia la derecha (es el único camino posible) y no puedes volver a las zonas que vayas dejando atrás.

Existen cuatro sectores, de dificultad creciente, que hay que superar para ganar la partida: Base de misiles, Puerto, Puente, y Campo de concentración.

Al comenzar, dispondrás de tres preciosas vidas. Podrás conseguir otra más a los 30.000 puntos, y luego otra por cada 70.000 adicionales.

Para jugar a **Green Beret,** no se requieren estrategias ni nada parecido. Es más, pensar demasiado puede resultar contraproducente. Todo es acción. Los enemigos aparecen constantemente al frente, por la espalda, o saltando sobre el sufrido «boina verde» desde alguna plataforma.

Lo mejor que podemos decir sobre lo bueno de este programa, es que tiene muy poco de malo. **Konami** nos está acostumbrando, desde bombas como el **Hypersports,** a programas de gran calidad que reúnen todas las condiciones que se le pueden exigir a un producto bien hecho. Los únicos defectos que hemos encontrado (y hemos perdido mucho tiempo buscándolos), son más achacables a las limitaciones gráficas del **Spectrum** que el propio programa.

DATOS GENERALES

TITULO Green Beret

FABRICANTE Konami

ORDENADOR Spectrum 48 K

TEMA DEL PROGRAMA Comando de élite

CALIFICACION (Sobre 10 ptos.)

ORIGINALIDAD	9
INTERES	8
GRAFICOS	8
COLOR	7
SONIDO	6
TOTAL	38





SUSTRAIDO UN MSX 2

El pasado mes fue sustraido del interior de un automóvil un ordenador Sony MSX 2, HB-F500P. Por ser un modelo aún poco difundido es facilmente identificable.
Se gratificará a quien de una pista que lleve a su recuperación. El número de serie es: 400286.
Llamar a la redacción de INPUT.

VISITANTES!

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16

S

Lab.

HANGAR

ZONA 5

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 14

S

Lab.

REACTOR NUCLEAR

ZONA 4

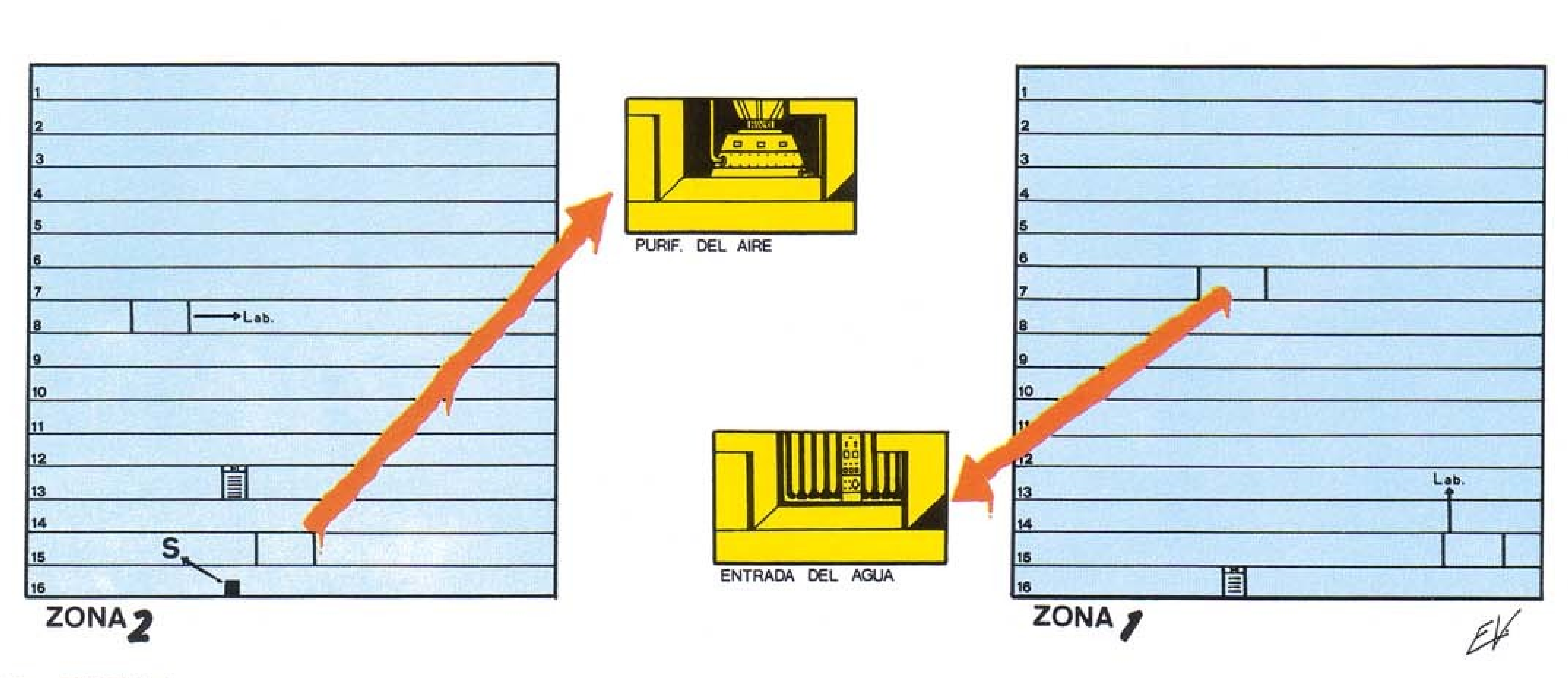
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16

Lab.

S

ORDENADOR CENTRAL

ZONA 3



Tanto por su título como por su tema, «V» puede parecer un programa oportunista más, uno de tantos. Sin embargo, y esperemos que la racha continúe, no es así. No recordamos —y tenemos buena memoria— ningún programa que, como éste, haya sido llevado del cine o la televisión al *soft* sin olvidar que dos mil pesetas bien merecen que el producto sea bueno.

TE LO CONTAMOS

La trama del juego se basa, cómo no, en el argumento de la popular serie de televisión que le da nombre. **Donovan,** el «bueno de la película», se ha infiltrado en la nave nodriza de los visitantes para destruirla. Va equipado con cinco potentes explosivos, que debe colocar en otros tantos puntos vitales de la nave, y activarlos mediante un mecanismo de relojería, teniendo en cuenta que debe sobrarle tiempo para escapar. Opcionalmente, puede reunir las cinco partes de la fórmula de un compuesto letal, el «polvo rojo», y esparcirlo por la nave a través de la planta de purificación del aire, acabando así con sus enemigos.

Aunque el nivel de dificultad no es muy alto, enfrentarse al programa sabiendo sólo lo poco que dicen las instrucciones puede resultar decepcionante. Por esta razón, haremos una descripción detallada del juego que pueda suplir, aunque solo sea en parte, este defecto.

La nave está dividida en cinco inmensas zonas, cada una de ellas con 16 niveles de pasillos. Para pasar de un pasillo a otro, **Donovan** debe estar colocado sobre uno de los discos en relieve que se encuentran en el suelo, cuidando que ninguna parte de su cuerpo haya quedado fuera. Pulsando «P» o «L», será teletransportado al nivel inmediatamente superior o inmediatamente inferior, respectivamente, siempre que en ellos también exista un disco en el mismo lugar.

Situándose detrás de una de las puertas negras que se encuentran en



los laterales de los pasillos, y pulsando «L», se acceda a la zona siguiente. Sólo hay una de estas puertas en cada zona, excepto en la

DATOS GENERALES

TITULO V

FABRICANTE Ocean

ORDENADOR Spectrum 48 K

TEMA DEL PROGRAMA
Ataque a nave invasora

CALIFICACION (Sobre 10 ptos.)

ORIGINALIDAD	8
INTERES	8
GRAFICOS	9
COLOR	9
SONIDO	3
TOTAL	37

última, que lógicamente no tiene. Las hemos señalado en el mapa con una «S».

Si lo que se quiere es pasar de una zona cualquiera a la anterior, hay que utilizar las puertas situadas en los laterales contrarios, las mismas por las que aparecerá **Donovan** cuando haga uso de una puerta negra para pasar a una nueva zona.

Para poder entrar en un laboratorio a recoger una parte de la fórmula, o en un punto vital de la nave para volarlo, es necesario descifrar primero un código. Una vez dentro, hay que actuar con la máxima rapidez, pues de lo contrario habrá que repetir de nuevo la operación para salir.

Donovan dispone en su misión de dos valiosísimas ayudas: una pistola láser y un mini-computador de múltiples usos, imprescindibles para enfrentarse con éxito a los peligrosos sistemas de seguridad de la nave.

Pulsando «L» (siempre que **Conovan** no se encuentre sobre un disco teletransportador), se accede al mini-computador. Este se maneja accionando doce teclas por medio de un cursor y el botón de disparo. Seis de ellas sólo sirven para trabajar descifrando claves e introducir los tiempos de detonación, y el resto para operar las siguientes funciones: coordenadas de posición, detener el juego, ver las partes de la fórmula que faltan, preparar una detonación, descifrar códigos, y volver a controlar a **Donovan.** Más adelante explicaremos con detalle el uso de algunas de estas funciones.

La pistola láser es de gran utilidad contra los robots que defienden la nave. Dichos robots pueden ser de cuatro tipos: los de mantenimiento, limpieza y vigilancia, que no matan pero ponen en peligro el corazón del héroe, y los de seguridad, de efectos más contundentes. Se puede seguir en todo momento el estado físico de **Donovan** atendiendo a la línea del electrocardiograma, que aparece en la parte inferior de la pantalla.

POR MAS SEÑAS...

«V» es un programa creado por **Kaos,** con la producción de **J. Woods** y la forma de la casa **Ocean.** La compañía **Warner Bros,** propietaria de los derechos originales de la serie de televisión, también se lleva una parte del pastel.

La comercialización del juego ha sido casi simultánea en Inglaterra y España, para variar, y se ha producido hace muy poquito tiempo.

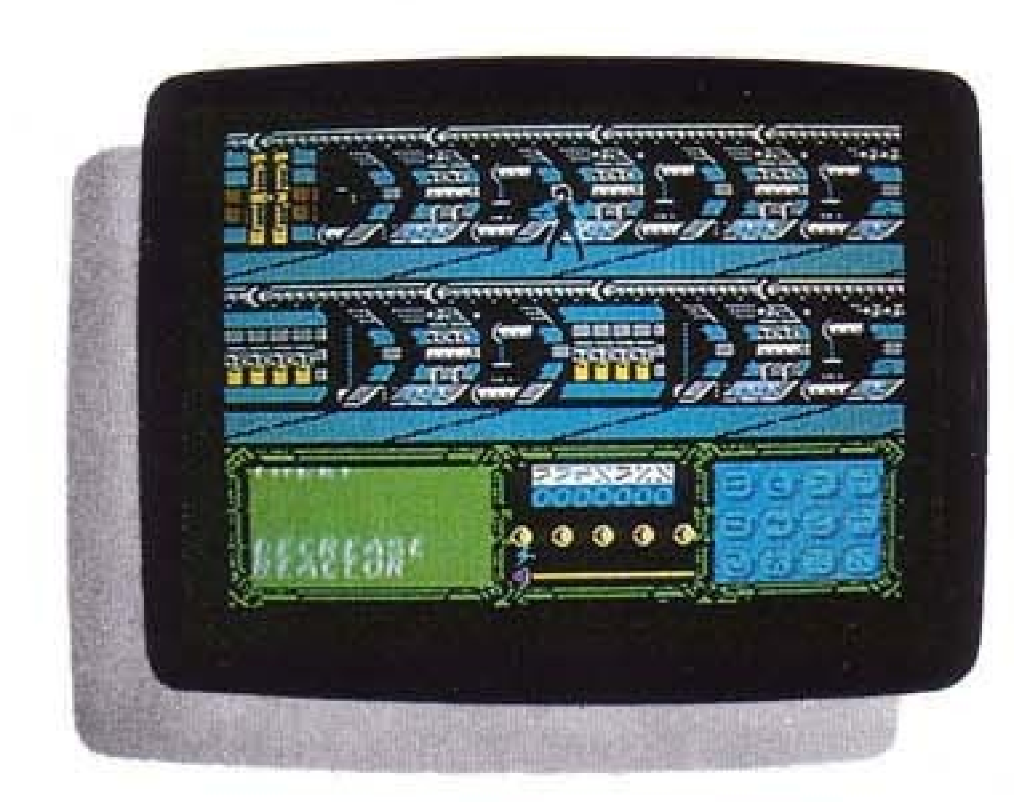
LO BUENO Y LO MALO

«V» es fundamentalmente un juego de acción en el que podemos

«pararnos a pensar». Más importante que nuestros reflejos, nuestra inteligencia tiene que trabajar a fondo descifrando códigos y orientándonos en el laberinto.
Afortunadamente, no se trata, como casi siempre, de andar matando a diestro y siniestro. Y esto, al menos en nuestra opinión, es bueno.
Aplaudimos sin reservas su originalidad, pues a pesar de estar basado en una serie de televisión de dudosa calidad, no presenta el típico y sospechoso parecido que suelen lucir casi todos los juegos, salvo honrrosas excepciones, con respecto a otros programas anteriores.
La buena factura y abundancia de los gráficos también son elementos dignos de mención, aunque no podemos dejar de señalar el hecho de que tal abundancia se debe a la utilización (sabiamente administrada pero, al fin y al cabo, nunca deseable) de una secuencia repetida de imágenes que discurren de forma similar a como ocurre en los «dibujos animados».
Un detalle que nos ha gustado mucho: el laberinto de pasillos no tiene siempre la misma estructura, y cambia completamente cada vez que jugamos. Esto impide que nos aprendamos un «recorrido» que nos lleve siempre a la victoria y que, irremediablemente, nos haga perder interés por el juego.
También nos ha impresionado favorablemente el logradísimo «teletransporte» dd **Donovan,** comparable a los mejores efectos especiales de la películas de ciencia-ficción.

LA VOZ DE SU AMO

Nos hubiera gustado ofrecer un mapa con todo lujo de detalles sobre el interior del laberinto. Pero como dicho interior cambia de una partida a otra, no ha dido posible hacerlo.
Afortunadamente, lo que no varía (pues de lo contrario se le pondría el asunto muy negro a nuestro héroe) es la localización de los puntos vitales de la nave, los laboratorios, las salidas y las entradas. Aunque el laberinto cambie, podréis comprobar que los lugares señalados en el mapa, que son los que nos interesan, siempre están en el mismo sitio.
La aventura comienza en la zona cinco. Cuando alcancemos el punto vital de esta zona, el hangar de atraque, nos encontraremos con dos problemas que se volverán a repetir a lo largo del juego:
¿Cómo descifrar la clave? ¿Cómo activar la bomba?





Contestando a lo primero, diremos que no hemos hallado ningún método infalible para conseguirlo rápidamente, pero sí vamos a explicar lo que las instrucciones del juego no dicen y debemos saber.
Colocando la flechita del computador sobre el símbolo situado en la parte inferior izquierda del mismo, aparece en la pantalla pequeña una sucesión de símbolos. Debemos conseguir que todos ellos sean iguales. Para lograrlo, hay que colocar la flecha sobre los signos que ocupan las dos columnas de la derecha del computador, iguales a los anteriores, y pulsar la tecla de disparo. De esta forma, según el signo elegido, variarán algunos caracteres de la sucesión. Con un poco de práctica y de paciencia, podremos conseguirlo sin mucho esfuerzo.
En cuanto a la activación de la bomba, debemos colocar la flecha en el segundo símbolo de la segunda columna, y pulsar el botón de disparo. En ese momento nos pedirá el computador, a través de la pantalla pequeña (y en inglés, naturalmente), que introduzcamos el tiempo para iniciar la cuenta atrás.
Pero no podremos elegir el que queramos, ni tampoco utilizar números convencionales. Usando de nuevo las dos columnas de la derecha, hay que introducir los tres signos que encontremos en ese momento en el reloj situado sobre el electrocardiograma y el contador del láser, empezando a contar a partir del segundo, para marcar después el segundo símbolo de la segunda columna (de nuevo). Si no hemos hecho correctamente la operación, el intento será abortado y deberemos intentarlo de nuevo.
Siempre que deseemos volver a controlar a **Donovan** y dejar el computador, colocaremos la flecha sobre el tercer símbolo de la segunda columna, y oprimiremos el botón de disparo.

EN RESUMEN

Un buen juego. Interesante, con las dosis justas de acción y estrategia, buenos gráficos y un colorido que no solemos ver en el **Spectrum.**

****

PING - PONG

Los juegos deportivos son de los programas que obtienen una mayor aceptación y en esta ocasión le toca el turno al ping-pong, un atractivo deporte que cuenta con muchos seguidores en los paises orientales.
En pantalla aparece el recinto de juego, con la mesa en perspectiva,

los espectadores en los laterales, el marcador en las esquinas inferiores y, por supuesto, las dos raquetas, que son movidas por manos invisibles.
Se pueden realizar diversos movimientos como mates, dejadas y lanzamientos a las esquinas, todo ello con la parte interior o exterior de la raqueta.
Cada cinco puntos se cambia de servicio y se dispone de siete segundos para realizar el mismo, perdiéndose un tanto si transcurre el tiempo sin lanzar la pelota. No hay un segundo servicio, por lo que si se falla el primero se pierde un punto.
Se juega el mejor de once tantos, pero siempre que la diferencia sea de dos o más puntos; en caso contrario continúa el juego hasta que uno de los dos consiga esa diferencia.
A medida que se vaya mejorando en el juego se pasará a un nivel superior, con un contrincante mucho más difícil. En total se presenta cinco niveles de dificultad, el último de los cuales convierte al **Spectrum** en un auténtico profesional de la raqueta pequeña, alcanzando un ritmo trepidante en cada tanto.
Las aficiones se sitúan en los laterales del campo y están claramente diferenciadas, de este modo los apasionados espectadores que presencian el *match* aplauden cada tanto de su jugador.
Es posible utilizar *joystick* y el partido se puede realizar entre dos jugadores o contra la máquina.
Un entretenido juego de **Konami** que cuenta con una extraordinaria presentación musical, con efectos sonoros muy bien conseguidos simulando una verdadera composición.





DATOS GENERALES

TITULO Ping - Pong

FABRICANTE Konami

ORDENADOR Spectrum 48K

TEMA DEL PROGRAMA Simulación deportiva

CALIFICACION (Sobre 10 ptos.)

ORIGINALIDAD	8
INTERES	7
GRAFICOS	7
COLOR	6
SONIDO	10
TOTAL	38

★★★★★★★★★★★★★★★★★★★★★★★★

NAPOLEON, SEÑOR DE LA GUERRA

La evidencia de que los juegos de guerra son menos comerciales que los de acción y aventuras, y la mala prensa que personas malinformadas han hecho contra ellos, han obligado a las compañías punteras en materia de *software* a eludir prudentemente este tema y a dejarlo en manos de firmas de segunda fila. Por esta razón, si ya es bastante difícil encontrar un programa que se presente como un *«wargame»*, resulta prácticamente imposible conseguir uno que verdaderamente lo sea.
Hasta el momento, exceptuando varios títulos de reciente aparición (entre ellos los que aquí comentamos), solo **Arnhem** podía considerarse como un buen *wargame*, entre cinco o seis títulos existentes en el mercado.
Afortunadamente, desde hace escasísimas fechas se está produciendo una mini-avalancha de programas de este tipo que podemos considerar como de un nivel aceptable. Entre ellos están **Austerlitz** y **Waterloo**, ambientados en dos de las grandes batallas de las guerras napoleónicas. Comentamos ambos simultáneamente por tratarse de dos juegos muy similares, apoyados en un mismo programa base y con las mismas reglas, aunque comercializados como dos programas por separado.
En ellos se aportan una serie de interesantes innovaciones con

respecto a las posibilidades que ofrecen los juegos de «fichas y tablero» tradicionales, como la opción que permite dar órdenes colectivas a un grupo de unidades, y la existencia de generales con voluntad propia al mando de una parte de las tropas, que pueden aconsejar al jugador o incluso ignorar sus órdenes y actuar «por su cuenta».

Se puede elegir entre tres niveles de dificultad, aunque advertimos que en el nivel dos resulta dificilísimo vencer, y en el tres prácticamente imposible.

DATOS GENERALES

TITULO Waterloo y Austerlizt

FABRICANTE Lothlorien

ORDENADOR Spectrum 48K

TEMA DEL PROGRAMA
Batallas napoleónicas

CALIFICACION (Sobre 10 ptos.)

ORIGINALIDAD	6
INTERES	8
GRAFICOS	4
COLOR	4
SONIDO	-
TOTAL	22





En los combates se utilizan tres criterios para determinar el nivel de fuerza de una unidad: los efectivos, la moral y el terreno.

Existe la posibilidad de informarnos sobre estos factores antes de realizar un ataque, de forma que podamos hacernos una idea de cuál va a ser el resultado y si nos conviene o no.

A pesar del interés que presentan ambos juegos, padecen de una serie de limitaciones que ya habían sido superadas en el pasado por otros *wargames* de menor calidad: no se puede elegir bando (el jugador lleva siempre a los franceses y el ordenador a los aliados); el escenario es de dimensiones reducidas y el repertorio gráfico pobre; hay que cargar de nuevo el programa para jugar una nueva partida; la artillería no aparece representada.

En aras a una mayor sencillez y fluidez del juego, se ha menoscabado una parte del rigor histórico y realístico, pero por ello no pierde interés, excepto para aquellos que pertenezcan a la sesuda élite de los «wargamenómanos».

★★★★★★★★★★★★★★★★★★★★★★★★★★

LA OLIMPIADA BLANCA

Año 1988 en Calgary, Alberta (Canadá) los juegos de invierno van a comenzar. Sobre un hermoso paisaje nevado y en el recinto del palacio de hielo, se van a celebrar las pruebas olímpicas.

Un total de seis pruebas han de ser realizadas, cada una de las cuales exige unas determinadas características.

La competición se inicia con el peligroso salto de trampolín, en el que no sólo cuenta los metros que se consigan alcanzar sino también el estilo con el que se realice el salto. Se dispone de tres intentos para obtener la máxima puntuación.





A continuación con un trampolín y un par de esquíes, hay que conseguir realizar bonitas filigranas que sean merecedoras de una alta puntuación. Un riguroso jurado compuesto por ocho personas, puntuará los movimientos desarrollados en cada uno de los tres intentos, otorgando un máximo de diez puntos.

La tercera prueba que completa la primera parte del juego es la velocidad sobre patines, donde durante 250 millas hay que deslizarse sobre el hielo con la mayor rapidez posible.

El esquí de fondo es una dura prueba que combina la fuerza física con la destreza. A lo largo de un difícil camino, hay que subir y bajar





C/ DUQUE DE SESTO, 50. 28009-MADRID
TELEF.: (91) 275 96 16
METRO O'DONNELL O GOYA
APARCAMIENTO GRATUITO EN FELIPE II

SOFTWARE: ¡¡2 PROGRAMAS POR EL PRECIO DE 1!!
Y ADEMAS, COMPLETAMENTE GRATIS, UN MAGNIFICO RELOJ DE CUARZO. INCREIBLE ¿VERDAD?

PING PONG	**2.295** ptas.
SABOTEUR	**2.295** ptas.
RAMBO	**2.295** ptas.
YIEAR KUNG FU	**2.295** ptas.
WORLD SERIES BASEBALL	**2.095** ptas.
MAPGAME	**2.750** ptas.
RAID	**2.295** ptas.
MIKIE	**2.100** ptas.
HIGHWAY ENCOUNTER	**1.750** ptas.
KNIGHT LORE	**1.750** ptas.
FAIRLIGHT	**1.950** ptas.
BATALLA DE LOS PLANETAS	**2.100** ptas.
ASTROCLONE	**1.900** ptas.
TURBO ESPRIT	**2.100** ptas.
DYNAMITE DAN	**2.100** ptas.
THE SOLD A MILLION	**2.500** ptas.
OLE TORO	**2.100** ptas.
MASTER OF T. LAMP	**1.950** ptas.
NIGHTSHADE	**1.750** ptas.
HACKER	**1.950** ptas.
SUPER TEST	**2.300** ptas.
ALIEN 8	**1.750** ptas.
GUNFRIGHT	**1.750** ptas.
TRES S. PARAISO	**2.100** ptas.
COSMIC WARTOAD	**2.100** ptas.
N.O.M.A.D.	**2.100** ptas.

SOFTWARE DE REGALO:
DUMMY RUN
YENTZ
DRAGONTORC
FIGHTING WARRIOR
MAPSNACH
SHADOW FIRE
SOUTHERN BELLE
POLE POSITION
BOUNTY BOB
SAIMAZOOM
TAPPER

SPECTRUM PLUS + 6 JUEGOS
¡¡**27.800** PTAS.!!
GRATIS: 2 WALKIE TALKIES!!

¡¡NUEVOS PROGRAMAS EXPLOSIVOS FUTURE STARS!!
ALI BEBE   KRIPTON RAIDERS   TOMMY
LOS TRES, **1.895** PTAS. Y ADEMÁS, GRATIS, 1 CALCULADORA

LAPIZ OPTICO DK'TRONICS
¡¡**2.890** PTAS.!!

INTERFACE-1 **10.900** PTAS.
MICRODRIVE **10.900** PTAS.

CASSETTE ESPECIAL PARA ORDENADOR **5.295** PTAS.

AMPLIACIONES DE MEMORIA
¡¡**3.995** PTAS.!!

SERVICIO TECNICO DE REPARACION
TARIFA FIJA: **3.600** PTAS.

OFERTA IMPRESORAS: TODAS LAS MARCAS CON UN ¡¡20% DTO. SOBRE P.V.P.!!

PRECIOS EXCEPCIONALES PARA TU AMSTRAD
CPC-464   CPC-6128 Y PCW-8256

TECLADOS PROFESIONALES:
— SAGA-1 **9.295** PTAS.   — INDESCOMP **13.195** PTAS.

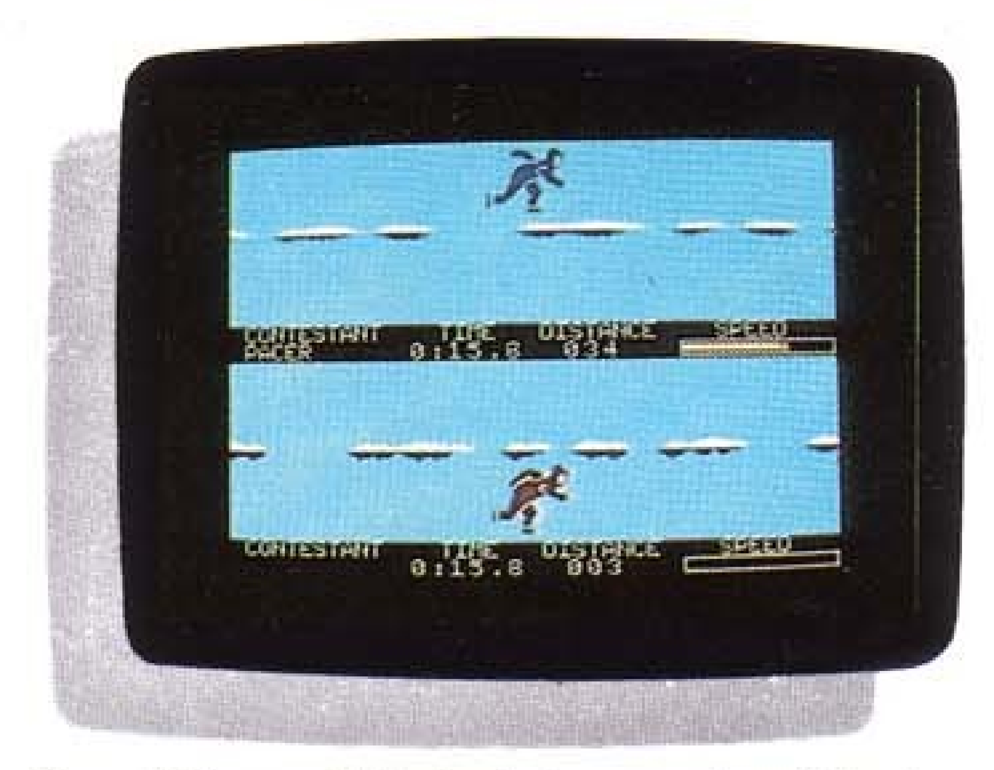
WAFADRIVE **16.995** PTAS.   QUICK DISK 2.8" **23.995** PTAS.

OFERTAS JOYSTICK QUICK SHOT
GRATIS   CALCULADORA
QUICK SHOT II + INTERFACE **3.695** PTAS.
QUICK SHOT V + INTERFACE **3.995** PTAS.

CARTUCHOS MICRODRIVE	**495** PTAS.
DISKETTES 5 1/4"	**295** PTAS.
DISKETTES 3"	**990** PTAS.
CARTUCHERAS PARA MICRODRIVE	**150** PTAS.
CINTA C-15 ESPECIAL ORDENADOR	**69** PTAS.
INTERFACE CENTRONICS/RS-232	**8.495** PTAS.
AMPLIFICADOR DE SONIDO	**2.390** PTAS.
INTERFACE DOBLE KEMPSTON	**2.795** PTAS.

Pedidos contra-reembolso sin ningún gasto de envío. Teléf.: (91) 275 96 16 o escribiendo a MICRO-1
C/ Duque de Sesto, 50. 28009-Madrid





DATOS GENERALES

TITULO Winter Games

FABRICANTE Epyx

ORDENADOR Spectrum 48K

TEMA DEL PROGRAMA
Simulación deportiva

CALIFICACION (Sobre 10 ptos.)

ORIGINALIDAD	7
INTERES	7
GRAFICOS	7
COLOR	6
SONIDO	6
TOTAL	**33**

montañas, y en cuatro puntos del recorrido es necesario disparar a unas dianas situadas a gran distancia. La rapidez con la que se realice la prueba y la puntería con el fusil determinarán los puntos correspondientes.
Para demostrar las habilidades personales nada mejor que el patinaje artístico sobre hielo. Una prueba de un minuto para realizar una serie de ejercicios obligatorios y dos minutos para crear las figuras más difíciles y hermosas. Ocho jueces dictaminan la puntuación merecida en función del estilo exhibido y los fallos tenidos.

Por último el *bobsled*, una trepidante bajada en trineo a través de un tobogán de hielo. Cuanto mayor es la velocidad, menor tiempo se empleará en el recorrido pero será más difícil de dirigir y mayor el riesgo de volcar.
En el juego pueden competir hasta cuatro jugadores, realizar sólo alguna de las pruebas o simplemente practicar, así como utilizar simultáneamente dos *joysticks*.
A este invernal juego de **Epyx** le acompaña una bonita melodía que aparece entre las distintas pruebas.

EL ZORRO CABALGA DE NUEVO

El malvado sargento **García** ha raptado a una hermosa señorita, haciéndola prisionera en su inexpugnable fortaleza.
El **Zorro**, misterioso personaje de novela, presencia el forcejeo, convirtiéndose en testigo directo del rapto. Desde este momento se inicia el asalto a la fortaleza con un claro objetivo: rescatar y devolver la libertad a la muchacha.
Por desgracia, los secuaces del sargento **García** intentarán atrapar al **Zorro** para conseguir la recompensa que ofrecen.
Todos aquellos que se atrevan a luchar con él pueden salir malparados y cuando sean derrotados, la marca del **Zorro** se reflejará en el aire.
Aunque pueda parecer que es posible alcanzar al raptor, es mejor no intentarlo ya que tras dura persecucción se introduce en el castillo y resulta imposible entrar en él.
Cuando el juego se inicia el **Zorro** aparece junto a un pozo, frente a una esquina de la fortaleza, momento en el que la dama deja caer su pañuelo. Existen dos caminos para alcanzar el escondrijo del sargento **García**. Uno pasa por el pueblo y atraviesa el cementerio. El otro es a través del pozo que se alcanza trepando por la rosaleda y saltando al agua.
Durante el transcurso del juego hay que ir recogiendo los objetos que parpadean, aunque sólo es posible transportarlos de uno en uno. Si en un momento dado parece imposible traspasar un obstáculo es conveniente saber qué objetos como







sofás, fuelles o flotadores pueden proporcionar el impulso necesario para superarlo.
Si se opta por el camino del pozo, algunas pantallas presentan auténticas pruebas de ingenio, con poleas, pesas, contrapesos, elevadores...
El programa de **U.S. Gold** presenta un sonido algo pobre, pero que es obviado por lo entretenido que resulta hacer «zetas» espadachinas en las panzas de nuestros malvados enemigos.

DATOS GENERALES

TITULO El Zorro

FABRICANTE U.S. Gold

ORDENADOR Spectrum 48K

TEMA DEL PROGRAMA
Aventuras del Zorro

CALIFICACION (Sobre 10 ptos.)

ORIGINALIDAD	7
INTERES	7
GRAFICOS	7
COLOR	6
SONIDO	6
TOTAL	33

****

LA LECCION DE ANATOMIA DEL PROFESOR SPECTRUM

Aunque existe una gran oferta de programas de entretenimiento, el **Spectrum,** como cualquier micro, tiene muchas más posibilidades. Las aplicaciones educativas son una de las más interesantes, ya que convierten al micro en un instrumento no sólo de distracción sino también en una herramienta que puede proporcionar una importante utilidad.
Las ventajas de la enseñanza asistida

por ordenador las podemos centrar en las posibilidades gráficas, la participación del estudiante, la constante repetición de la lección, la realización de pruebas de control y el

DATOS GENERALES

TITULO El cuerpo humano

FABRICANTE Tasoft

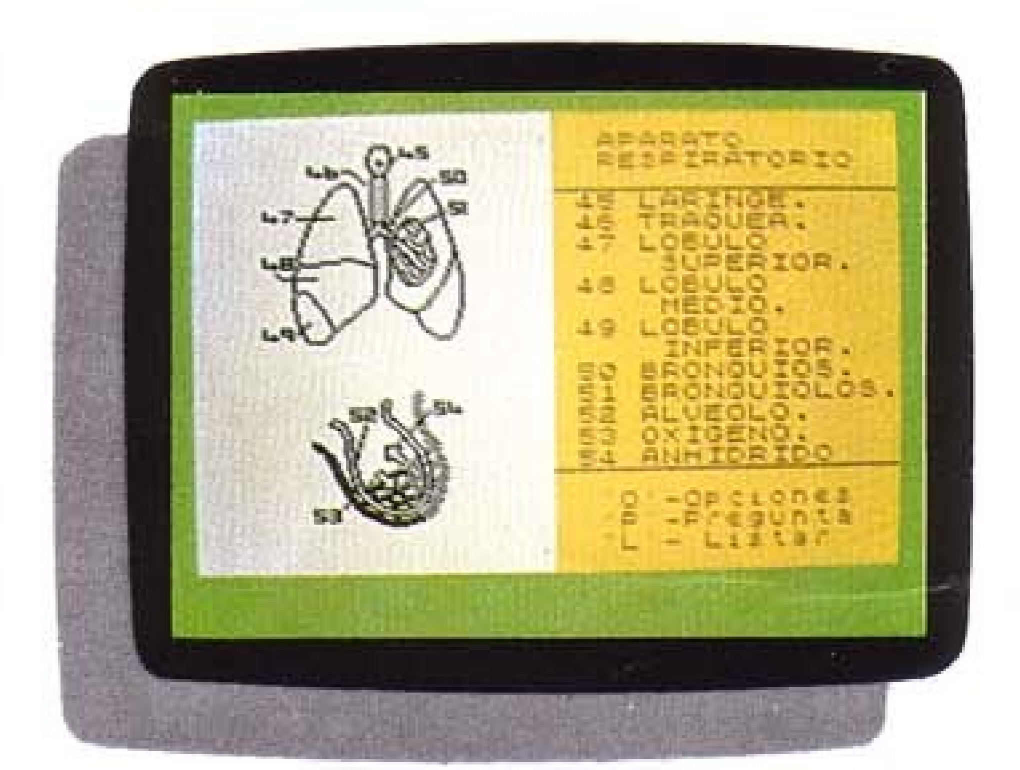
ORDENADOR Spectrum 48 K

TEMA DEL PROGRAMA Educativo

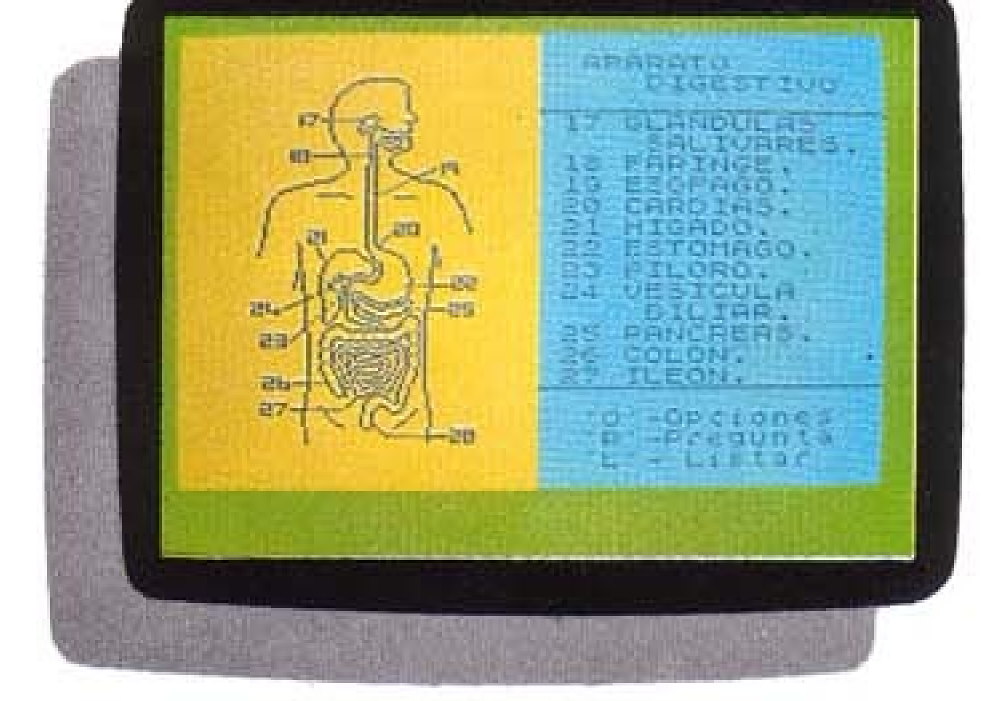
CALIFICACION (Sobre 10 ptos.)

ORIGINALIDAD	8
INTERES	9
GRAFICOS	9
COLOR	6
SONIDO	—
TOTAL	32









indudable atractivo que supone el micro.

El programa que aquí presentamos, **«El cuerpo humano»,** es una creación de **Tasoft** que tiene como objetivo enseñar una lección de anatomía con un incansable profesor y un participativo alumno.

El juego incluye dos *cassettes,* uno relativo al esqueleto y otro a las funciones vitales del cuerpo humano, ambos totalmente independientes.

El esqueleto humano ofrece un menú inicial con cinco opciones: catálogo del esqueleto, catálogo por campos, guía del esqueleto y los *test* 1 y 2.

El catálogo de esqueleto contiene los nombres de un total de 51 huesos que pueden observarse en un gráfico al margen. El catálogo por grupos diferencia el esqueleto, cabeza, columna, tórax, mano y pie, repartiendo los 51 huesos entre estos seis grupos. La guía del esqueleto permite preguntar por el hueso que

SUSTRAIDO UN MSX 2

El pasado mes fue sustraido del interior de un automóvil un ordenador **Sony MSX 2, HB-F500P.** Por ser un modelo aun poco difundido es facilmente identificable.
Se gratificará a quien de una pista que lleve a su recuperación. El numero de serie es:400286.
Llamar a la redacción de INPUT.

deseamos conocer, destacándolo en negro en el esqueleto o en la ampliación, lo cual permite definir totalmente el tamaño y la posición de los 51 huesos considerados. Por último la opción *test* 1 y 2 ofrecen la posibilidad de pasar un pequeño examen que refleje los conocimientos adquiridos, calculando el porcentaje de aciertos. El *test* 2 sirve para realizar una competición entre dos estudiantes, preguntándose alternativamente por alguno de los huesos.
El segundo *cassette* analiza las funciones vitales de nuestro organismo, con un menú de diez opciones, que son: constitución general, partes principales, aparato digestivo, circulatorio, respiratorio y excretor, esquema del encéfalo y los *test* 1 y 2.
La constitución general diferencia las células, tejidos, órganos, sistemas y aparatos. Las funciones vitales informan sobre la nutrición, relación y reproducción que a su vez tienen varias subdivisiones auto explicativas.
Las partes principales muestran un esquema del cuerpo humano con un total de dieciséis secciones numeradas, así como su posición y forma dentro del organismo.
Los siguientes apartados analizan los principales procesos del cuerpo humano, la digestión de alimentos, la circulación de la sangre, la respiración, el aparato excretor y el cerebro.
El aparato digestivo muestra el dibujo de un cuerpo humano con las doce partes principales que componen el proceso digestivo.
Para explicar el funcionamiento de la circulación de la sangre aparece en pantalla un esquema completo del aparato circulatorio con sus principales arterias que confluyen en los vasos capilares y un dibujo de la pieza fundamental del engranaje, el corazón, con sus aurículas y ventrículos. Un total de dieciséis partes componen este apartado.
El aparato respiratorio tiene en los pulmones su principal protagonista, mostrándose una sección del mismo, incluyendo también un dibujo de los alveolos, con el mecanismo de funcionamiento.
Un esquema del aparato excretor, y los riñones como piezas fundamentales, constituyen otra de las opciones que presenta este programa, diferenciándolo un total de catorce partes vitales.
Por último, el menú presenta dos esquemas del cerebro humano con catorce secciones que ofrecen una aproximación al más complejo de nuestros organismos.
El programa se completa con la posibilidad de realizar un examen de los conocimientos adquiridos, contestando a las 82 posibles preguntas o partes del organismo diferenciadas.
En definitiva un programa educativo con unos gráficos excelentes que muestra, una vez más, las posibilidades de nuestro microordenador.

★★★★★★★★★★★★★★★★★★★★★★★★★★

EL ESPIRITU DE RASPUTIN

Nos encontramos en la antesala del lado oscuro del **Mundo** de los **Siete Planetas.** Nuestra misión: neutralizar los ocho hechizos de *Rasputín* y destruir su fuente de energía.
Para ello, debemos derrotar a las malvadas criaturas que se interpongan en nuestro camino, sortear un gran número de obstáculos, y recuperar las fuerzas que vayamos perdiendo en el intento. Disponemos de la única ayuda de una espada de energía, un escudo, y naturalmente, nuestra habilidad a los mandos del **Spectrum.**
La acción transcurre en una serie de pantallas de desarrollo tridimensional que, aunque se parecen mucho entre sí, siempre encierran peligros diferentes.
El nivel de dificultad es quizás demasiado alto, pero esto, lejos de restar interés al programa,



contribuye a conseguir que no nos podamos separar de la pantalla hasta haber logrado algún pequeño éxito. No obstante, es muy probable que nos desesperemos un poco al principio. Es cuestión de voluntad.
Al final de cada partida, podremos seguir nuestros progresos comprobando el nivel alcanzado en la tasa de «credibilidad del heroe», expresada en tantos por ciento, y

DATOS GENERALES

TITULO Rasputín

FABRICANTE Firebird

ORDENADOR Spectrum 48 K

TEMA DEL PROGRAMA
Aventura épica

CALIFICACION (Sobre 10 ptos.)	
ORIGINALIDAD	7
INTERES	7
GRAFICOS	8
COLOR	4
SONIDO	5
TOTAL	31





que al principio no subirá del 00,00%.
Los gráficos y el planteamiento general del programa están muy inspirados en la línea *Ultimate* (abierta con los famosos *Alien-8* y *Night-Lore*), aunque no podemos decir que les falte originalidad.
El único defecto que hemos encontrado a este interesante juego es la monotonía cromática de sus pantallas. Todas ellas son siempre de un solo color, hecho que los autores disculpan poniéndolo en relación con el propio argumento de la aventura, pero que a nosotros no nos ha gustado mucho. Es un detalle que desluce considerablemente la presentación del programa. Pero a pesar de ello, el balance general es muy positivo.

★★★★★★★★★★★★★★★★★★★★★★★★★★

VIVA ESPAÑA!!

DATOS GENERALES

TITULO Costa Capers

FABRICANTE Firebird

ORDENADOR Spectrum 48 K

TEMA DEL PROGRAMA Aventura turística

CALIFICACION (Sobre 10 ptos.)

ORIGINALIDAD	5
INTERES	6
GRAFICOS	7
COLOR	5
SONIDO	4
TOTAL	**27**

Costa Capers, fabricado por la firma **Firebird,** es un juego bastante poco original que podemos clasificar dentro del más genuino estilo **Manic-Miner.** A pesar de tratarse de un modelo muy repetido, que ya empieza a aburrir a algunos usuarios, no podemos negar que sí presenta un aspecto verdaderamente novedoso (novedoso y poco afortunado), como es su tema argumental: el autor ha tenido la genial idea de utilizar la típica y absurda imagen que se tiene de España fuera de nuestras fronteras para animar su programa y darle un toque «diferente» (recordando el famoso *slogan* turístico). No faltan en el juego toros, soles, sombrillas ni castañuelas, es decir, todo lo necesario para que cualquier turista que se precie se sienta como en su casa.
La aventura consiste en guiar a un pequeño (pequeñísimo) personaje llamado **Ted** en su recorrido turístico por España, sorteando multitud de peligros. Su objetivo es permanecer borracho todo el tiempo, tener cuidado con las insolaciones, conseguir dinero para comprar cosas que le sean útiles y, sobre todo, sacar 36 fotografías de otros tantos lugares pintorescos, y llevarlas a revelar antes de que terminen sus bien merecidas vacaciones. Todo ello amenizado, cómo no, con las notas de la popular canción de **Manolo Escobar** «Que viva España».





A pesar de lo desafortunado del tema, el programa está bastante bien realizado y entusiasmará a los forofos «manicmineros». Las únicas notas negativas, desde el punto de vista técnico, son la poca espectacularidad de los gráficos y las pequeñas dimensiones del protagonista, que es de un tamaño aproximado de dos caracteres.
No podemos dejar de llamar la atención del lector sobre el sistema de carga empleado, que resulta muy original (aunque no es la primera vez que se emplea), y puede sorprender un poco al principio. Las típicas rayitas transversales que bailan en los márgenes cuando estamos cargando un programa, desaparecen en este caso sin que por ello se produzca ningún tipo de error. Mientras tanto, van apareciendo en la pantalla las instrucciones y un reloj de cuenta-atrás en el que podemos ver el tiempo que falta para que concluya la carga.



EL ZOCO

Intercambio programas de Spectrum 48K. Mandar lista a:

Olga Bulcao Asensio
Conjunto Europa Edf. Oslo 7°A 2
Torre del Mar (Málaga)

Desearía intercambiar programas comerciales para Spectrum. Mandar lista de programas a:

Ricardo Estévez Serrano
Ricardo Zamora, 3 (TRAFICO)
30003 Murcia

Compro cualquier programa en el que se pueda jugar al futbol y vendo por 1000 ptas. «El libro gigante de los juegos para ZX Spectrum», con unos 150 juegos. También puedo cambiar el libro por el juego, tengo un Spectrum Plus.

Iko
Eskalantegui, 41, 1°D
Tel. 51 81 24
Paxai Antxo 20000 Guipuzcoa

Cambio, compro, vendo programas, libros, ideas, etc. para el Spectrum 48K. Compro interface a ser posible doble y joystick en buen estado y a buen precio, no importa que esté usado.

Jesús López Martinez
Diego de Torres, 7, 2°A
Tel. 889 37 84
Alcalá de Henares (Madrid)

Vendo cassette marca Computone por 3.800 ptas. Escribir a:

Agatha Estera
Calabria, 281, 4°F
Tel. 239 89 23
Barcelona

Cambio por Spectrum 48K, consola de video-juegos Atari 2600, con todos los cables, 2 joysticks y 3 juegos, E.T. Pinball, Maze Craze. En buenas condiciones. LLamar:

Jorge Fernández
Tel. (93) 562 21 41
Barcelona

Por cambio de ordenador estoy interesado en deshacerme de mis programas para Spectrum. Escribid a:

Juan Carlos Hernández
Condado de Treviño, 75, 2°D
Miranda de Ebro (Burgos)

Intercambio programas de juegos, utilidades, educativos, etc. para ordenador ZX Spectrum de 48K. Interesados enviar lista de programas a:

José Ramón Román Collado
Avda. de la Raza, 1, D 7°
41012 Sevilla

Vendo Spectrum 16K con 4 meses de garantía y una colección de 59 juegos por 9.000 ptas (negociables). Interesados llamar a:

Mauro Javier Oruezabal Moreno
Avda. Oroel, 26
Tel. 36 04 57
Huesca

Vendo ZX Spectrum 48K con cables, manuales (español) y el embalaje, con más de 30 juegos y revistas por 19.500 ptas. comprado el 27-8-85. También ZX Interface 2, con un cartucho de ROM y joystick Quick-Shot II por 2.500 y 1.500 ptas. Todo por 23.000 ptas. También lo cambio por MSX

Francisco Noriega
Cipriano Pedrosa, 15 bajo izda.
Tel. (985) 69 58 82
Sama de Langreo (Asturias)

Vendo Spectrum Plus nuevo con más de 50 juegos comerciales. Una suscripción a INPUR SINCLAIR por un año y muchas revistas y libros de informática (Spectrum). Todo 30.000 ptas.

Marc Crespell
Miguel Utrillo, bloque 15
Urbanización Baipeneda
Tel. 894 26 78
Sitges (Barcelona)

Ocasión 20.000 ptas. Vendo ZX-Spectrum 48K, fuente de alimentación, juego de cables, manuales y cinta de demostración por poseer dos. Perfecto estado, pocas horas de uso.

P.J. González
Aptdo. de Correos 1415
La Laguna (S.C. de Tenerife)

Desearía contactar con usuarios del Spectrum para el intercambio de ideas, mapas, programas y software en general; también estoy interesado en la compra y venta de lo anterior.

Emerio Ares Senra
Avda. Coruña 360, 8°F
27003 Lugo

Vendo ordenador ZX-81 con ampliación de memoria 16K, en perfecto estado. Precio 10.000 ptas. Diregirse a :

José Isaac Aranda Jiménez
Condesa Pardo Bazán, 22 Sobreático 1ª
Tel. 349 90 82
08027 Barcelona

Deseo intercambiar programas con otros usuarios de Spectrum. Interesados mandar lista de programas a la siguiente dirección:

Gorka Eizagirre
Avda. Zarauz, 68, 3° izda.
Tel. 21 34 82
San Sebastián (Guipuzcoa)

Se vende ZX-81 recién estrenado por 5.000 ptas, o se cambia por cassette propio de computadora en buenas condiciones. Interesados escribir a:

Oscar Cabezas Fariñas
Iparraguirre, 73, 12°
Tel. 431 39 27
48012 Bilbao

Cambio Atari con dos clases de mandos, conmutador de imágenes y tres juegos (Combat, Defender, Comecocos) por ordenador Spectrum 48K.

Victor
Tel. (943) 36 42 27
Llamar de 8.30 a 9.30

Vendo joystick e interface, perfecto estado, por el precio de 3.500 ptas. Interesados escriban a:

Ramón Guillén Barrera
San Carlos, 138 atl/B
Tel. (965) 21 02 14
(de 6 a 8 de la tarde)
03013 Alicante

Vendo ordenador Spectravideo SV-318 nuevo (dic.85) por 30.000 ptas. También vendo para el Spectrum las cintas originales de: «10 juegos II» de Microparadise y el «Hiperspots», por 1.000 ptas. y 1.500 ptas. respectivamente. Las dos por 2.000 ptas. Interesados escribir a:

Hair Soft
Palamios, 5, 2°D
Pontevedra

Cambio libros y programas «ZX-Spectrum» por libros y/o programas. Interesados escribir a:

Luis Ortiz Cruz
Antonio Machado, D-8, 1° 1ª
08033 Barcelona

Deseo cambiar juegos para Spectrum 48K, con usuarios de Málaga. Tengo buenos títulos: Spy Hunter, Skool Daze...

Javi
Tel. 32 28 26
Málaga



SISTEMAS EXPERTOS. CONCEPTOS Y EJEMPLOS

Autor: J.L. Alty y M.J. Coombs
Editor: Díaz de Santos
Páginas: 218
Precio: 2.120 ptas.

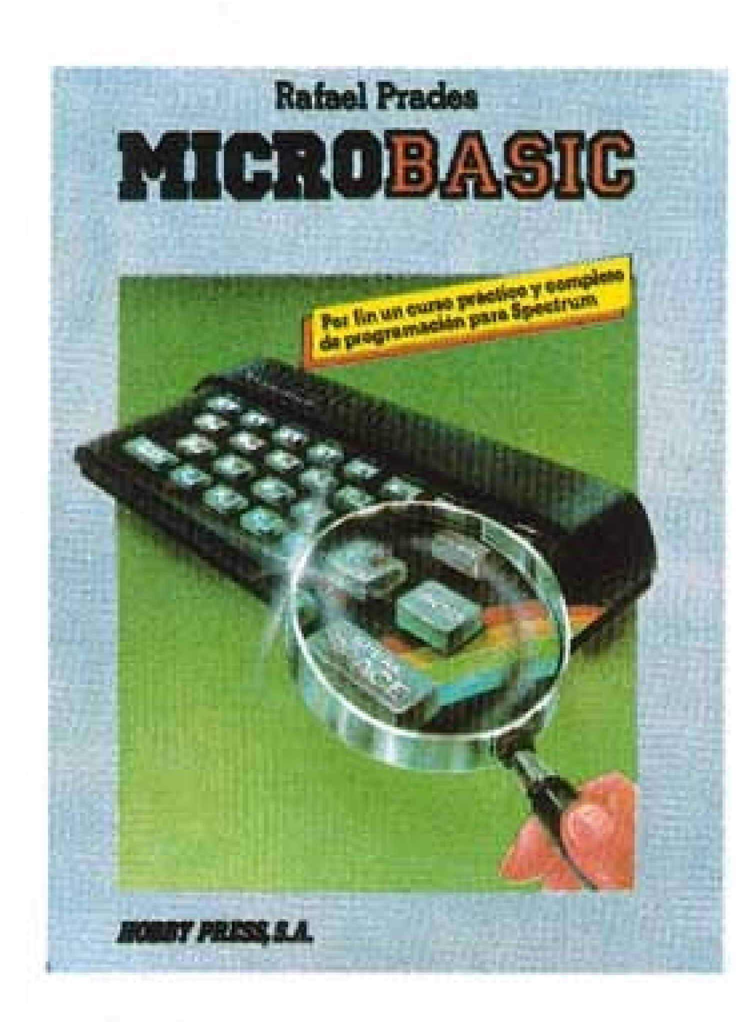
Para aquellos que están muy interesados en el mundo de la **Inteligencia Artificial** y los **Sistemas Expertos**, a condición de que ya posean un buen nivel de conocimientos y muchas ganas de aprender, recomendamos especialmente un libro de reciente aparición que promete satisfacer la curiosidad de los más avezados devoradores de bibliografía. Advertimos que su contenido no es de aplicación concreta en ningún ordenador específico, sino que se trata de una descripción exhaustiva de los principios generales de los **S.E.** a un nivel profundo.

Los autores comienzan la obra comparando el proceso de datos tradicional con los **Sistemas Expertos**, para pasar seguidamente a describir con detalle los métodos de representación y control, la «inferencia lógica», el cálculo de predicados, y su aplicación para la resolución de problemas.

Nos hablan también del ya tópico sistema de «factorización» para la reducción del tiempo empleado en buscar un dato en una tabla, de interesante aplicación en los ficheros.

Desgraciadamente, una de las partes más interesantes se refiere a diversos sistemas existentes en el mercado, como el famoso «PROSPECTOR», que no están al alcance de nuestro bolsillo y mucho menos de nuestro «micro» (y de los que no son tan «micros»). No obstante, bien es cierto que no es posible felicitarnos por nuestros conocimientos sobre **Inteligencia Artificial** sin haber oído hablar de dichos sistemas.

La obra termina, como no, con una panorámica de los nuevos desarrollos en materia de **Sistemas Expertos** muy actualizada (aunque sepamos que en un par de años se quedará obsoleta), especialmente dedicada a aquellos que siempre quieren estar al cabo de «lo último».



MICRO BASIC

Autor: Rafael Prades
Editor: Hobby Press
Páginas: 334
Precio: 1.750 ptas.

Para aquellos que el manual del Spectrum se les haya quedado pequeño y quieran ampliar sus conocimientos sobre este micro, el libro que aquí presentamos puede significar una importante ayuda para profundizar en las posibilidades que brinda esta máquina.

A lo largo de más de 300 páginas se van analizando los aspectos más importantes del micro de una forma clara y sencilla, con el único requisito de tener un Spectrum, ya que no se precisan conocimientos previos para iniciar su lectura.

Tras la introducción se realiza una presentación del equipo, con unos consejos sobre los cuidados y mantenimiento de los medios de almacenamiento. También incluye unas nociones sobre los sistemas binario, decimal y exponencial, así como el uso de los operadores matemáticos y lógicos.

La primera parte finaliza con una tabla del código ASCII y un amplio ejemplo del desarrollo de un programa con su organigrama correspondiente.

La segunda parte está dedicada a analizar detalladamente todas y cada uno de los comandos y sentencias que ofrece el Spectrum. De este modo con explicaciones y ejemplos se puede conocer con bastante exactitud las funciones y utilidades del Basic de Sinclair. Comenzando por los comandos básicos y de control, cada capítulo analiza un aspecto distinto, como son los bucles, las subrutinas y la introducción de datos y su lectura.

Otros capítulos están dedicados a las funciones, las cadenas, las matrices y la depuración de programas, ofreciendo un gran número de ejemplos que contribuyen a una mejor comprensión de los comandos.

El color, los gráficos y el sonido cuentan con su capítulo específico, así como las sentencias de grabación y carga, con ejemplos y pequeñas recetas.

También son objeto de análisis las impresoras, el interface I, la memoria y los periféricos, completando el libro un apéndice que incluye el código ASCII y el juego de caracteres del Spectrum.

Se trata de un atractivo libro lleno de color e ilustraciones que junto con un buen número de esquemas conforman lo que podemos considerar como un completo manual. Además cuenta con casi ochenta programas de ejemplo y múltiples recetas que contribuyen a hacer más amena y digerible su lectura.

CODIGOS Y CLAVES SECRETAS

Autor: Gareth Greenwood
Editor: Anaya
Páginas: 308
Precio: 1.378 ptas.

No es preciso ser un avezado agente secreto, curtido en las redes del espionaje internacional, para sentir cierta curiosidad por la codificación y el desciframiento de mensajes. Sin embargo, hasta ahora era preciso disponer de cierta formación para tener entrada en esta compleja y apasionante disciplina. En adelante, este libro ofrece multitud de ideas y pautas, capaces de iniciar a cualquier usuario de ordenador en el apasionante juego de complicar la existencia de los intrusos de la información o, por otro lado, desgranar los misterios ocultos tras una nota cifrada. La más importante ventaja de la obra consiste en poner la criptografía al alcance del BASIC.

Se describen métodos tan simples como la clásica permutación de las letras del alfabeto, hasta los diferentes tipos de transposición, sin olvidar los cifrados **Gronsfeld**, **Bifido** o **Hill**.

Un loable apéndice se ocupa de las peculiaridades ofrecidas por nuestro idioma, particularización clave en los métodos que recurren al análisis de la frecuencia de la presencia de las letras, que presentan un perfil diferente del idioma en que fue escrito el libro.

La profusión de listados de impresora convierte a **Códigos y Claves Secretas** en una herramienta práctica, que huye de la agobiante teoría.

Por último una nota, aunque se utiliza un BASIC bastante estandarizado, los listados han sido adaptados para ser tecleados en buen número de ordenadores de gran difusión. Este es otro ejemplo del buen hacer de la colección **Anaya Multimedia**.

